8-2009

Optimization Models for Designing Spatially Compact Ecological Reserve Systems

Lakmali Weerasena
Clemson University, lweeras@clemson.edu

Optimization Models for Designing Spatially Compact Ecological Reserve Systems

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Masters in Science
Mathematics

---

by
Lakmali Weerasena
August 2009

Accepted by:
Dr. Douglas Shier, Committee Chair
Dr. Herve Kerivin
Dr. David Tonkyn
Dr. Matthew Saltzman

Abstract

Over the past decades, a number of mathematical models and solution techniques have been developed to preserve reserve sites for species and their natural habitats. Two optimization models for designing spatially compact ecological reserve systems are addressed here as zero-one integer programming problems. These formulations have a bicriteria objective function that is a combination of both boundary length and distance. The two formulations cluster the sites into a relatively small number of compact groups while preserving a required number of sites that contain a certain species using a given amount of resources. Two general types of approaches have been developed to solve the resulting mathematical models: exact and heuristic algorithms.

Acknowledgments

I would like to express my heartiest gratitude to my advisor Dr. Douglas Shier for his guidance and support. His thoughtfulness and new ideas inspired me and helped me immensely in making this a success. My sincere thanks are due to Dr. Herve Kerivin, for sharing his experience with us about integer programming. Also my many thanks go to Dr. David Tonkyn for sharing his biological experience and ideas with us to complete this successfully, and to Dr. Matthew Saltzman for advising me with selection of computational tools and finally but not last to my husband and all of my friends who helped me to complete this successfully.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Methods for designing reserve systems for ecological species have been considered in several papers in the past decade. In general, the problem statement is to find a subset of reserve sites that minimizes the cost of establishing reserve sites containing a given set of species or that maximizes the number of species present under a given budget constraint. Both of these types of reserve site selection problems can be formulated as integer programming (IP) problems and represented as either a set covering problem (SCP) or a maximal covering problem (MCP).

In the set covering formulation, given a set of target species and a set of potential sites, we wish to determine the least-cost reserve system that satisfies a specified minimum representation for each species. In the maximal covering formulation, with a limited conservation budget, the objective is to determine a reserve system that includes the maximum number of species. These aspects have been considered in several papers [12, 17].

In practice reserve design problems need to consider more than just species coverage and budget limitations. Other spatial characteristics such as the distance between selected reserve sites and the shape of the reserve system should be considered as well. Several mathematical models that consider spatial optimization have been proposed to address the important issues of representation of species within reserve systems [9, 10]. Such approaches make it possible to design a better spatial arrangement for a reserve system by considering attributes such as contiguity and the shape of the selected sites.

Because the SCP and MCP formulations do not consider spatial relationships between the sites selected for the reserve system, the resulting reserve system may be highly fragmented. The

meaning of fragmentation depends on the specific objectives of the conservation programs. For example, if a reserve system consists of many small habitat areas, it may not facilitate the movement of species between habitat areas. In this way small disconnected reserve systems may be harmful to the survival of the species within the reserve. Moreover the contiguity of a reserve may be important to species survival within the reserve. For example, a contiguous reserve system helps species to roam freely within the system without leaving the space.

More compact reserve systems help to reduce the edge effects of the system such as invasion of predators. Also compact reserve systems help to improve the buffering effects by absorbing the disturbances and other adverse impacts. A variety of shape measures have been proposed in reserve selection models to represent compactness. Some of the proposed measures are the boundary length of the reserve, the ratio of boundary length to area, and the average distance between the sites in the reserve system.

Therefore contiguity and compactness can be important in modeling reserve site selection problems. A variety of formulations have been proposed to address these contiguity and compactness attributes. Some formulations have explicitly considered both contiguity and compactness together (using a linear combination) while others have considered only one attribute. Now we briefly discuss some of these studies.

Shirabe [18] proposed a new exact formulation for structural contiguity that can be incorporated into any mixed integer programming model. According to this model, the resulting system enforces contiguity regardless of the other included criteria such as compactness. Graph theory approaches have also been proposed to control the contiguity of reserve systems. Onal and Briers [14] developed a linear IP formulation that uses a graph theory approach to obtain a connected reserve system. Although this formulation ensures contiguity, it contains some sites named as gap sites that are to be excluded in the final solution. Therefore their objective was to minimize the total number of gap sites. They used additional variables and constraints explicitly to avoid cycle formation. Onal and Wang [15] developed an improved linear IP formulation, also using a graph theory approach. Their objective was again to minimize the total number of gap sites. The main difference between these two formulation is the method used to avoid cycle formation. Although the model in [14] explicitly uses additional constraints and variables to avoid cycles, the improved model [15] does not. Rather, if cycles are present in the solution, new cuts are added and the model is solved again. The authors of the improved model [15] mentioned that their model is computationally more efficient

because of its reduced size. Both of these formulations focus on the structural contiguity of a reserve system. Hof and Flather [7] developed a different nonlinear IP model that preserves the contiguity of the system by controlling the shape, requiring reserves to be either circular or rectangular.

Several mathematical models have been proposed to group disconnected sites together into compact reserves. A collection of adjacent reserve sites is called a *cluster*. In these models [4, 5, 9, 10, 11, 13] such reserves of compact shapes are generated as clusters. In an ecological sense clusters might correspond to as different habitats. Separated clusters (habitats) may be desirable because such a separation will preserve the species in the face of natural disasters such as destruction of the habitat by fire. Also clustering adjacent sites improves the opportunities for multiple biological interactions among the given species.

Onal and Briers [13] developed two integer programming approaches to the problem of reserve selection to obtain compact reserve systems. In the first approach, they minimized the sum of the distances between all pairs of sites included in the reserve system. In the second approach, an alternative formulation minimizes the largest distance between selected sites instead of the total distance. Fischer and Church [4] presented a linear IP formulation for minimization of the boundary length to promote reserve aggregation and compactness.

Fischer and Church [5] developed a bi-objective formulation by considering both the boundary length and the site selection cost. McDonnell et al. [9] developed a bi-objective nonlinear IP formulation which involves a weighted combination of the boundary length of the selected clusters and the area of selected sites. They mentioned that minimizing the area of the selected sites is equivalent to minimizing the cost of the selected sites. Nalle et al. [10, 11] developed a nonlinear formulation which explicitly addresses the compactness and shape of the selected reserve sites. This model minimizes a weighted combination of two measures: the boundary length of selected clusters and the distance between all pairs of selected sites (even those in disjoint clusters).

The purpose of the current study is to develop a bi-objective mathematical model for selecting reserve sites which clusters the sites into a relatively small number of compact groups. The main difference between this new formulation and those given in [10, 11] is the way we measure distance between selected sites. We measure the distances within each cluster, whereas the models in [10, 11] measure the distances between all selected sites whether such sites are in the same cluster or different clusters. In some situations consideration of distance within clusters rather than the total distance between all sites may be more useful. Since each cluster might be treated as a different habitat, in

general it is not important to consider the distance between different habitats. Typically, there are reduced biological interactions among geographically separated clusters. For example if one habitat represents a mountain and the other habitat represents a swamp, there is no need to measure the distance between these two habitats to obtain compact clusters. Thus if an optimum distance can be maintained between all the sites within a given cluster, this will assure maximum interactions among different sites within the cluster. Therefore it is important to minimize the distance within clusters in order to produce compact clusters.

When creating a cluster the boundary length of the selected cluster is especially important. For example, Figure 1.1(a) and Figure 1.1(b) illustrate two possible clusters each containing four sites. Assuming that each site is a unit square, the cluster in Figure 1.1(a) has ten boundary edges while the cluster in Figure 1.1(b) has eight boundary edges. Since each site is a unit square, the number of boundary edges is equivalent to the overall boundary length. Visually it is clear that the cluster in Figure 1.1(b) is more compact than the cluster in Figure 1.1(a). Therefore when compact clusters are desired, it seems reasonable to pick clusters having a small boundary length relative to the area. From an economic point of view also, this is important since the cost of management often depends upon the boundary length of the reserve area, with longer boundaries requiring higher maintenance cost.

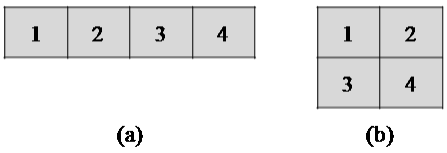

Figure 1.1: Two clusters on four sites with different boundary lengths

In our models we therefore treat boundary length as our primary objective. But simply minimizing the boundary length does not ensure compact clusters for the system. To illustrate, Figure 1.2 shows two clusters, each containing three (unit square) sites and each having the same boundary length eight.

Let us consider the Euclidean distance between the centers of each site in the cluster. The sum of the Euclidean distances between distinct sites in Figure 1.2(a) is 4 while the corresponding sum in Figure 1.2(b) is 3.41. Visually it is clear that the cluster in Figure 1.2(b) is more compact

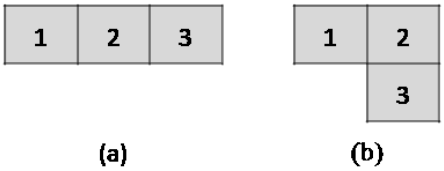

Figure 1.2: Two clusters on three sites with the same boundary length

than that in Figure 1.2(a). Therefore it seems reasonable to select among clusters with the same boundary length one having the smaller sum of (Euclidean) distances. This distance measure thus provides a secondary criterion. As suggested by this example, our strategy for creating a compact cluster involves two steps. We first consider the boundary length of the cluster and among all such clusters of minimum boundary length we pick a cluster that minimizes the distance between all pairs of sites within the cluster.

In general we consider these two aspects in designing a compact reserve system containing *several* clusters: minimizing the boundary length of all clusters and minimizing the total distances between all pairs of sites within each cluster. This should lead to a more compact reserve system. Thus we formulate the reserve design problem using an appropriate bi-objective optimization model.

To minimize the hierarchical combination of boundary length and then total within cluster distance, we use a technique from multi-objective programming [3] that weights these two objectives. The combining weight U is specified in a particular manner to give priority to minimizing the boundary length as the primary objective. We calculate this weight U by considering the l_1 distances between all possible pairs of sites in the reserve. Details of this calculation are explained in Section 2.2.

Now the objective function of our mathematical model is to minimize U times the boundary length of all clusters plus the sum of the Euclidean distances between sites within the designated clusters. This formulation also incorporates the requirements to cover all target species with a limited conservation budget.

Our initial formulation is a nonlinear integer programming model and therefore solving the model exactly is time consuming. In order to solve the model more efficiently, we convert the proposed model into a linear integer program. Details of this conversion are discussed in Chapter 2.

In Chapter 3 we provide numerical examples to compare computational aspects of the two models discussed in Chapter 2. Additional data sets are then used to explore more fully the computational behavior of our linear integer programming model. In Chapter 4 we present a heuristic algorithm to obtain good feasible solutions to our linear integer model for problems that are too challenging to solve exactly. Chapter 5 summarizes our work and discusses future research directions.

Chapter 2

Mathematical Models

For a given conservation area, we are interested in protecting certain species occurring within the region. It is assumed that the prevalence of existing species does not change with time. Also, the entire conservation area is considered to be partitioned into a number of potentials sites. Two sites are said to be adjacent if they share a common boundary.

The model developed below assumes that the study region is partitioned into uniformly sized sites for simplicity. The region is considered to be an $n \times m$ grid of uniform sites and each site is a 1×1 unit square. For example Figure 2.1 shows a 5×6 grid, containing 30 unit sites. The shaded site shows the site designated $(2, 3)$ in this grid; there are four other sites adjacent to site (2,3).

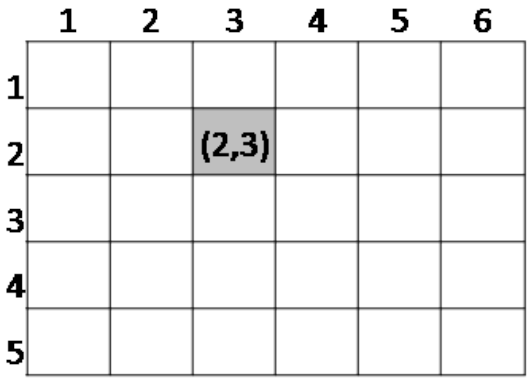

Figure 2.1: An example 5×6 grid

Let V denote the set of sites (or nodes) in the reserve system and let E denote the set of edges (adjacencies) in the system. In the given $n \times m$ grid, any node $v \in V$ can be written as (i, j) where $i = 1, \ldots, n$ and $j = 1, \ldots, m$. If two nodes $v_1 = (i, j), v_2 = (k, l) \in V$ are adjacent (i.e., their corresponding sites share a boundary) then the ordered pair $(v_1, v_2) = ((i, j), (k, l)) \in E$. For a grid system, each site can be adjacent to at most four other sites. The notation used in the mathematical model is described below.

S = total number of conservation species present in the reserve system

C = maximum number of possible clusters

A_s = set of sites inhabited by species of type s where $s = 1, 2, \ldots, S$

n_s = required number of sites for species of type s where $s = 1, 2, \ldots, S$

$N(i, j) = \{(k, l) \in V : ((i, j), (k, l)) \in E\}$, the set of nodes adjacent to node (i, j)

$D(i, j) = \{(k, l) \in V :$ either $(k \geq i$ and $l > j)$ or $(k > i$ and $l \leq j)\}$, See Figure 2.2

$d_{ij,kl}$ = Euclidean distance between the center of site (i, j) and the center of site (k, l)

b_{ij} = budgetary cost of conserving or purchasing site $(i, j) \in V$

B = total budget available for the reserve system

U = upper bound on the total Euclidean distance between all possible pairs of grid sites

The decision variables indicate which sites are included in the reserve and their allocation to clusters:

$$X_{cij} = \begin{cases} 1 & \text{if site } (i, j) \text{ is included in Cluster } c; \\ 0 & \text{otherwise.} \end{cases}$$

In this model, Cluster 1 denotes the sites *not* selected for conservation and the remaining clusters contain those sites that are selected for conservation. The clusters $c = 2, 3, \ldots, C$ are called *real* clusters since they are the ones containing the protected species. Because Cluster 1 is not selected for conservation, we actually have at most $(C - 1)$ real clusters.

The 0-1 quadratic optimization model can be written as follows:

$$(P_1)\ \text{Minimize} \sum_{c=2}^{C} \sum_{(i,j)\in V} \sum_{(k,l)\in D(i,j)} d_{ij,kl} X_{cij} X_{ckl} + U \times \left[\sum_{(i,j)\in V} \sum_{(k,l)\in N(i,j)} X_{1ij}(1 - X_{1kl}) \right]$$

$$\text{Subject to} \sum_{c=2}^{C} \sum_{(i,j)\in A_s} X_{cij} \geq n_s \quad \text{for all } s = 1, 2, \ldots, S \tag{2.1}$$

$$\sum_{c=2}^{C} \sum_{(i,j)\in V} b_{ij} X_{cij} \leq B \tag{2.2}$$

$$X_{c_1 ij} + X_{c_2 kl} \leq 1 \tag{2.3}$$

$$\text{for all } (c_1 \neq c_2) \text{ and } ((i,j),(k,l)) \in E \text{ and } c_1, c_2 = 2, 3, \ldots, C$$

$$\sum_{c=1}^{C} X_{cij} = 1 \ \text{ for all } (i,j) \in V \tag{2.4}$$

$$X_{cij} \in \{0, 1\} \ \text{ for all } c = 1, 2, 3, \ldots, C \text{ and } (i,j) \in V$$

2.1 Explanation of the optimization model

Let us consider each part of model (P_1) in detail.

1. Let us first discuss the objective function. As mentioned earlier the objective function consists of two parts. It is the weighted sum of

   (a) the distances between sites within the same real clusters and

   (b) the boundary length of all real clusters.

   (a) The first summation in the objective function calculates the total Euclidean distance between sites $(i,j) \in V$ and $(k,l) \in D(i,j)$ within the real clusters. Here the set $D(i,j)$ contains the distinct sites that occur "after" site (i,j) in a left-right, top-bottom ordering.

As depicted in Figure 2.2, the shaded area shows the distinct sites occurring after site (i, j).

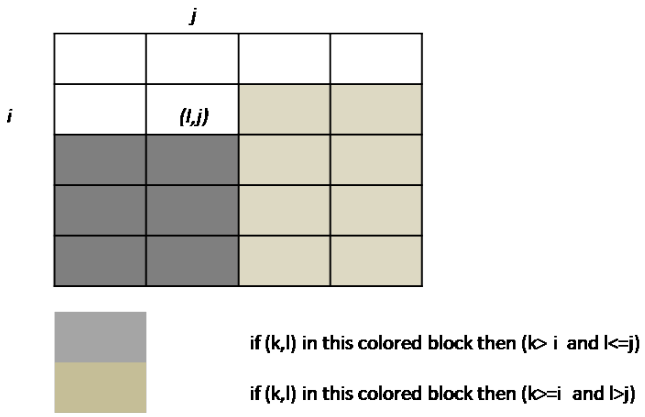

Figure 2.2: Explanation of the range $D(i, j)$

(b) The second summation gives the total number of boundary edges of all real clusters and thus the boundary length of all selected sites. To explain this summation, consider Figure 2.3(a) illustrating a site (i, j) with four neighbors.

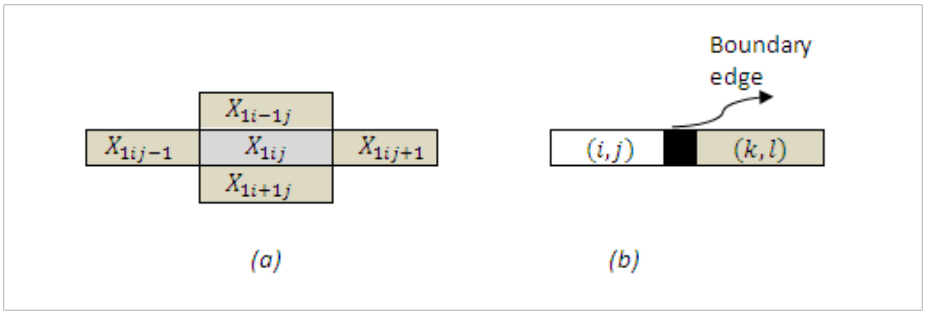

Figure 2.3: Explanation of the calculation of boundary length

If both (i, j) and (k, l) are selected for Cluster 1, then $X_{1ij} = X_{1kl} = 1$ and the product

$$X_{1ij}(1 - X_{1kl}) = 0$$

so no edge is counted towards the total boundary length. If (i, j) is in Cluster 1 and (k, l)

is in a real cluster (as shown in Figure 2.3(b)), then the product

$$X_{1ij}(1 - X_{1kl}) = 1$$

since $X_{1kl} = 0$ and thus a 1 is counted towards the total number of boundary edges. That is, since a real cluster is surrounded by Cluster 1, an edge is counted in the product above precisely when (i, j) is in Cluster 1 and (k, l) is in a real cluster. Therefore the second summation of the objective function gives the total number of boundary edges of all selected real clusters. Counting the total number of boundary edges then gives the boundary length of all clusters since we assume uniformly sized sites.

In the formulation of model (P_1), it is convenient to place a border of Cluster 1 sites surrounding the reserve system. This ensures that the outermost reserve sites are all adjacent to Cluster 1. Therefore if we have an $n \times m$ grid we are actually modeling an $(n-2) \times (m-2)$ reserve system.

2. Constraint (2.1) is the species representation requirement which states that to protect species of type s adequately, we must select at least n_s sites in which species s is present. Here we can note that only the selected reserve sites contribute to the species' representation and of course we consider only real clusters $(c > 1)$ for this purpose.

3. Constraint (2.2) states that the total cost of selected sites in the real clusters cannot exceed the conservation budget.

4. Constraint (2.3) enforces that if reserve sites $(i, j), (k, l)$ are included in two *different* real clusters then they do not share a boundary. This ensures that real clusters are disjoint from one another.

5. Constraint (2.4) states that each site is assigned to exactly one cluster $c \geq 1$.

2.2 Calculation of the upper bound U

As explained in the introduction we want to give priority to minimizing the boundary length. To do this we weight the total boundary length by a sufficient large value U. We now explain the

calculation of a suitable weight U.

One way to define a suitable weight is by using the total Euclidean distance between all possible pairs of sites $(i,j) \in V$ and $(k,l) \in D(i,j)$ in the reserve. First let us look at the Euclidean distance between pairs represented by an $n \times m$ grid of uniform sites. Let $d_{ij,kl}$ be the Euclidean distance between the center of site (i,j) and the center of site (k,l), where $(i,j) \in V$ and $(k,l) \in D(i,j)$. Then the total Euclidean distance between all pairs of sites is given by

$$D_2 = \sum_{(i,j)\in V} \sum_{(k,l)\in D(i,j)} d_{ij,kl}.$$

Because calculation of the sum D_2 is time consuming, we explore an easily calculated upper bound. Let $d_{11,nm}$ be the Euclidean distance between the center of site $(1,1)$ and the center of site (n,m). Assuming that the grid consists of 1×1 uniform sites, then

$$d_{11,nm} = \sqrt{(1-n)^2 + (1-m)^2}.$$

Clearly $d_{11,nm}$ is an upper bound on the Euclidean distance between any other two sites in the $n \times m$ grid. Since there are nm sites in the grid, an upper bound U' on the total Euclidean distance between all possible pairs of distinct sites in the grid is

$$U' = \binom{nm}{2} \sqrt{(1-n)^2 + (1-m)^2} = (1/2)nm(nm-1)\sqrt{(1-n)^2 + (1-m)^2}. \tag{2.5}$$

Instead of using the Euclidean distance l_2, we can consider the l_1 distance and obtain a better upper bound than U' on the total distance between all possible pairs of sites in the grid and one that can be calculated easily. The l_1 distance between two pairs of sites $(i,j) \in V$ and $(k,l) \in D(i,j)$ is defined as $d^1_{ij,kl} = |i-k| + |j-l|$. This is the rectilinear distance between sites (i,j) and (k,l). Then the total l_1 distance between all pairs of sites is given by

$$D_1 = \sum_{(i,j)\in V} \sum_{(k,l)\in D(i,j)} d^1_{ij,kl.}$$

Graovac and Pisanski [6] give a formula to calculate D_1, which is called the Wiener index of an $n \times m$ grid. According to this formula the total l_1 distance between all pairs of distinct sites

in the grid is given by

$$D_1 = \frac{nm(n+m)(nm-1)}{6}.$$

Now consider the ratio between the exact value of D_1 given by the Wiener formula and U', the upper bound (2.5) on D_2. This ratio is equal to

$$\frac{(n+m)}{3\sqrt{(1-n)^2+(1-m)^2}}.$$

It can be shown that the denominator of this fraction is larger than the numerator. Thus we have $D_1 \leq U'$.

Now we obtain a relation between D_1 and D_2 as follows. Let $A \equiv (x_1, y_1), B \equiv (x_2, y_2)$ be two points in R^2. Then the l_2 distance between A and B is $||AB||_2 = \sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$ and the l_1 distance between A and B is $||AB||_1 = |x_1-x_2|+|y_1-y_2|$. Consider Figure 2.4(a), in which the center of the unit circle and the rotated square is denoted O.

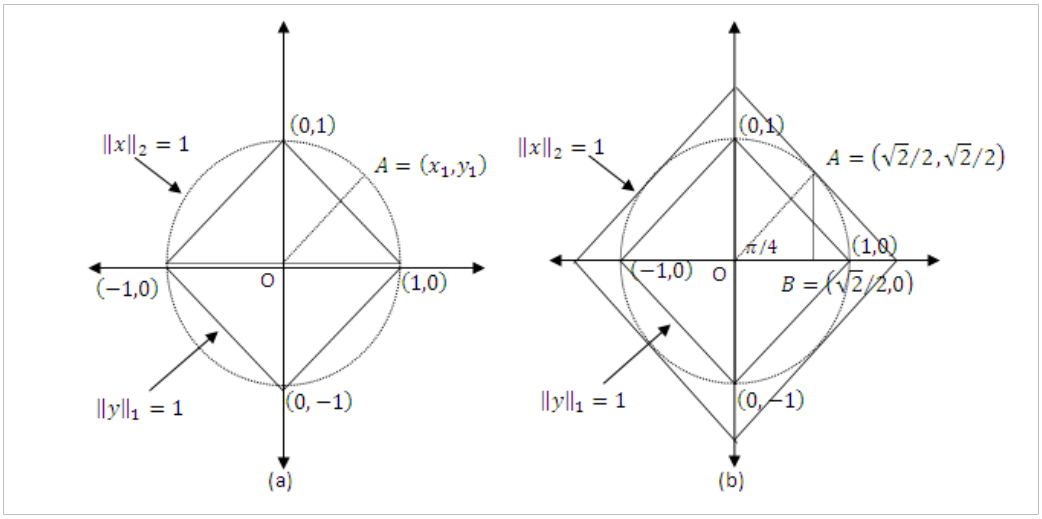

Figure 2.4: The relationship between the 1-norm and 2-norm

If we take any point x on the unit circle, the Euclidean distance from the origin to x is $||x||_2 = 1$. Similarly if we take any point y on the rotated square the l_1 distance from the origin to y is $||y||_1 = 1$.

As seen in Figure 2.4(a), it is clear that $||x||_2 \leq ||x||_1$. But if we pick the point $A \equiv (x_1, y_1)$ on the unit circle then $||OA||_2 = \sqrt{x_1^2+y_1^2}$ and $||OA||_1 = |x_1|+|y_1|$. It follows that $||OA||_2 \leq ||OA||_1$

always holds.

As we gradually increase the size of the inner rotated square in Figure 2.4(a), we get Figure 2.4(b) in which the outer rotated square touches the unit circle. This occurs when the angle between segment OA and the x axis is $\pi/4$. Then we have

$$||OA||_1 = \sqrt{2}/2 + \sqrt{2}/2 = \sqrt{2}$$

and

$$||OA||_2 = \sqrt{(\sqrt{2}/2)^2 + (\sqrt{2}/2)^2} = 1$$

so that

$$||OA||_1 = \sqrt{2}(||OA||_2).$$

For any other angle α where $0 \leq \alpha \leq \pi/4$, according to Figure 2.4(b) we have $\sqrt{2}||OA||_2 \geq ||OA||_1$. Since this figure is symmetric for any angle α where $0 \leq \alpha \leq 2\pi$ we have $\sqrt{2}||OA||_2 \geq ||OA||_1$.

By combining these results we have

$$||OA||_1/\sqrt{2} \leq ||OA||_2 \leq ||OA||_1.$$

More generally for any $A \in R^2$ we have $||OA||_1/\sqrt{2} \leq ||OA||_2 \leq ||OA||_1$ which implies that

$$D_1/\sqrt{2} \leq D_2 \leq D_1.$$

Since $D_1 \leq U$ from the previous discussion we obtain

$$D_1/\sqrt{2} \leq D_2 \leq D_1 \leq U'.$$

This shows that although U' provides an upper bound on D_2 we have a better set of bounds on D_2 as $D_2 \in (1/\sqrt{2}D_1, D_1)$ which can be calculated easily. Therefore we weight the total boundary length by using the calculated value of D_1. Table 2.1 shows values of D_1, D_2, U' and $D_1/\sqrt{2}$ for a 10×10 grid. It can be seen that the value D_1 is a reasonable upper bound on the value D_2 and

much better than the crude upper bound U'.

n	m	$\frac{D_1}{\sqrt{2}}$	D_2	D_1	U'
10	10	23334.5	25934.00	33000	63033.2

Table 2.1: Values of D_1, D_2, U' and $D_1/\sqrt{2}$ for a 10×10 grid

For ease of interpretation we take our upper bound U to be

$$U = 10^{\lceil \log D_1 \rceil} \tag{2.6}$$

where $\lceil \log D_1 \rceil$ indicates that the value $\log_{10} D_1$ is rounded to the next largest integer. The significance of using the modification (2.6) is that when the objective function value is displayed, the first nonzero digits correspond to the boundary length of the selected clusters followed by some zeros and then the total distance within clusters.

2.3 The size of model (P_1)

We first consider the total number of variables in model (P_1). Since the reserve system is represented by an $n \times m$ grid with nm sites and we permit at most C clusters, we have a total of nmC binary variables X_{cij}.

There are S constraints of type (2.1) because we have S species and also there is a single budget constraint (2.2). The inequality (2.3) ensures that real clusters are not adjacent. Although we have an $n \times m$ grid, we actually model an $(n-2) \times (m-2)$ grid. Thus we have $(n-3) \times (m-2)$ common horizontal interior edges shared by the sites and $(n-2) \times (m-3)$ common vertical interior edges shared by the sites. Therefore inequality (2.3) generates

$$(C-1)(C-2)\left[(n-3) \times (m-2) + (n-2) \times (m-3)\right]$$

$$= (C-1)(C-2) \times [2nm - 5(m+n) + 12]$$

constraints. Finally inequality (2.4) generates nm constraints for the system. In total we have

$$[S + 1 + (C-1)(C-2)\{2nm - 5(m+n) + 12\} + nm]$$

or $O(nmC^2)$ constraints for model (P_1).

2.4 A linear integer programming model

The objective function for model (P_1) is a quadratic function with nmC binary variables. As mentioned earlier it has two components: the sum of distances between sites within real clusters and the boundary length of the real clusters. Clearly the second component is not a convex function of the variables X_{cij}. In order to model the problem more effectively, we convert all quadratic terms of the objective function into linear terms by replacing each quadratic term $X_{cij}X_{ckl}$ by a new binary variable Y_{cijkl} :

$$Y_{cijkl} = \begin{cases} 1 & \text{if both } (i,j),(k,l) \in V \text{ are assigned to Cluster } c; \\ 0 & \text{otherwise.} \end{cases}$$

The following constraints ensure that Y_{cijkl} equals one if and only if sites (i,j) and (k,l) are both selected for Cluster c:

$$Y_{cijkl} \leq X_{cij} \text{ for all } (i,j),(k,l) \in V \text{ and } c = 1,2,\ldots,C \tag{2.7}$$

$$Y_{cijkl} \leq X_{ckl} \text{ for all } (i,j),(k,l) \in V \text{ and } c = 1,2,\ldots,C \tag{2.8}$$

$$X_{cij} + X_{ckl} - Y_{cijkl} \leq 1 \text{ for all } c = 1,2,\ldots,C,\ (i,j),(k,l) \in V \tag{2.9}$$

$$Y_{cijkl} \geq 0 \text{ for all } (i,j),(k,l) \in V \text{ and } c = 1,2,\ldots,C \tag{2.10}$$

1. Consider constraint (2.7) and constraint (2.8). These two constraints ensure that Y_{cijkl} must equal zero unless both X_{cij} and X_{ckl} equal 1.

2. Constraint (2.9) ensures that if both (i, j) and (k, l) are selected for Cluster c, then Y_{cijkl} must have value one.

The constraint set (2.7)–(2.9) is summarized in Table 2.2 which shows the possible values for the X and Y variables. Thus inequalities (2.7)–(2.9) ensure that $Y_{cijkl} = X_{cij}X_{ckl}$ always holds and the new mathematical model does not change the optimal solution of the original quadratic mathematical model (P_1).

X_{cij}	X_{ckl}	Y_{cijkl}
0	0	0
1	0	0
0	1	0
1	1	1

Table 2.2: Explanation of the constraints (2.7)–(2.9)

The new integer programming model can be represented as (P_2):

$$(P_2) \quad \text{Minimize} \sum_{c=2}^{C} \sum_{(i,j)\in V} \sum_{(k,l)\in D(i,j)} d_{ij,kl} Y_{cijkl} + U \times \left[\sum_{(i,j)\in V} \sum_{(k,l)\in N(i,j)} X_{1ij} - Y_{1ijkl} \right]$$

$$\text{Subject to} \sum_{c=2}^{C} \sum_{(i,j)\in A_s} X_{cij} \geq n_s \quad \text{for all } s = 1, 2, \ldots, S \tag{2.11}$$

$$\sum_{c=2}^{C} \sum_{(i,j)\in V} b_{ij} X_{cij} \leq B \tag{2.12}$$

$$X_{c_1ij} + X_{c_2kl} \leq 1 \tag{2.13}$$

$$\text{for all } (c_1 \neq c_2) \text{ and } ((i, j), (k, l)) \in E \text{ and } c_1, c_2 = 2, 3, \ldots, C$$

$$\sum_{c=1}^{C} X_{cij} = 1 \quad \text{for all } (i, j) \in V \tag{2.14}$$

$$Y_{cijkl} \leq X_{cij} \text{ for all } (i,j) \in V, (k,l) \in N(i,j) \text{ and } c = 1 \tag{2.15}$$

$$Y_{cijkl} \leq X_{ckl} \text{ for all } (i,j) \in V, (k,l) \in N(i,j) \text{ and } c = 1 \tag{2.16}$$

$$X_{cij} + X_{ckl} - Y_{cijkl} \leq 1 \text{ for all } c = 2,3,\ldots,C,\ (i,j) \in V,\ (k,l) \in D(i,j) \tag{2.17}$$

$$X_{cij} \in \{0,1\} \text{ for all } c = 1,2,3,\ldots,C,\ (i,j) \in V$$

$$Y_{cijkl} \geq 0 \text{ for all } c = 2,3,\ldots,C,\ (i,j) \in V,\ (k,l) \in D(i,j)$$

$$Y_{cijkl} \geq 0 \text{ for all } (i,j) \in V, (k,l) \in N(i,j) \text{ and } c = 1$$

Consider constraints (2.14) and (2.15). As explained earlier, $Y_{cijkl} \leq X_{cij}$, $Y_{cijkl} \leq X_{ckl}$ and $Y_{cijkl} \geq 0$ ensure that Y_{cijkl} must equal zero unless both (i,j) and (k,l) are selected. The first term in the objective function of model (P_2) contains Y_{cijkl} for $c > 1$ with a positive weight $d_{ij,kl}$. Thus in this case the objective function forces Y_{cijkl} to be as small as feasibly possible in order to minimize the objective function value. Therefore it is unnecessary to define (2.14)–(2.15) explicitly for real clusters. The second term in the objective function contains $-Y_{cijkl}$ with $c = 1$. In this case the negative sign of the objective function forces all Y_{cijkl} to be as large as feasibly possible in order to achieve a minimum value. Since constraint (2.16) ensures that the value of the Y_{cijkl} variable is equal to one when both sites (i,j) and (k,l) are selected, it is unnecessary to define constraint (2.16) explicitly for Cluster 1.

We measure the distance within real clusters in order to minimize the total distance. Thus to measure the distance between two sites (i,j) and (k,l), we need to consider $(i,j) \in V$ and $(k,l) \in D(i,j)$. Therefore when $c > 1$, constraint (2.16) only needs to be defined for $(i,j) \in V$ and $(k,l) \in D(i,j)$. To measure the boundary length of a real cluster, we need to consider a site $(i,j) \in V$ and a site $(k,l) \in N(i,j)$. Thus constraints (2.14)–(2.15) only need to be defined when $c = 1$.

Based on the above factors we define the Y variables for two index sets as ($Y_{cijkl} \in \{0,1\}$ for all $c = 2,3,\ldots,C,\ (i,j) \in V,\ (k,l) \in D(i,j)$) and ($Y_{cijkl} \in \{0,1\}$ for all $(i,j) \in V, (k,l) \in N(i,j)$ and $c = 1$). However the nature of the objective function and constraints (2.14)–(2.16) allows

us to relax the Y variables to be continuous variables: namely $Y_{cijkl} \geq 0$.

2.5 The size of model (P_2)

We now determine the total number of variables and constraints in the new model (P_2). First consider the objective function. As in the quadratic model this objective function contains two terms. The first term involves the distance between sites in the same real cluster and the second term involves the boundary length of each real cluster.

Consider the first term, which contains the variables Y_{cijkl} defined for real clusters $(c > 1)$ of the objective function. Since the first term of the objective function measures the distance between each $(i, j) \in V$ and each $(k, l) \in D(i, j)$ for all real clusters, it generates

$$\binom{(n-2)(m-2)}{2}(C-1) = (C-1)(n-2)(m-2)\left\{(n-2)(m-2)-1\right\}/2 \tag{2.18}$$

Y variables. Although we have an $n \times m$ grid, as mentioned earlier, we actually model an $(n-2) \times (m-2)$ grid and have at most $(n-2)(m-2)$ sites for real clusters. Thus it is not necessary to define variables Y_{cijkl} for all n and m since only interior sites are in real clusters. Therefore, out of the $(n-2)(m-2)$ X variables we choose two at a time to define Y variables for each real cluster.

The second term of the objective function measures the boundary length by counting the number of boundary edges for each real cluster as explained in model (P_1). To calculate the total number of boundary edges we need to consider all vertical and horizontal common interior edges. Since the total number of interior edges is equal to $(n-3)(m-2)+(n-2)(m-3)$, this part generates

$$2\left\{(n-3)(m-2)+(n-2)(m-3)\right\}$$

Y variables, as there are two Y variables associated with each interior edge.

As before the number of binary X_{cij} variables is nmC. In addition the objective function of model (P_2) contains

$$(C-1)(n-2)(m-2)\left\{(n-2)(m-2)-1\right\}/2 + 2\left\{(n-3)(m-2)+(n-2)(m-3)\right\}$$

or $O(n^2m^2C)$ continuous variables.

Now we consider the total number of constraints of the new linear model (P_2). Constraints $(2.10)-(2.13)$ in model (P_2) are exactly the same constraints as $(2.1)-(2.4)$ in the original quadratic model (P_1). Therefore, as explained in Section 2.3, these produce

$$S + 1 + (C-1)(C-2)\{2nm - 5(m+n) + 12\} + nm$$

or $O(nmC^2)$ constraints.

In addition, model (P_2) contains three types of constraints (2.14)–(2.16). Constraints (2.14) and (2.15) are only needed for Cluster 1 and for $(k,l) \in N(i,j)$. Since the number of interior edges is equal to $(n-3)(m-2) + (n-2)(m-3)$, constraints (2.14)–(2.15) together generate

$$4\{(n-3)(m-2) + (n-2)(m-3)\}$$

constraints. Constraint (2.16) is needed only for real clusters and for $(k,l) \in D(i,j)$. Since there are $(C-1)$ real clusters and $\binom{(n-2)(m-2)}{2}$ pairs of distinct sites, the number of constraints generated by (2.16) equals

$$(C-1)(n-2)(m-2)\{(n-2)(m-2)-1\}/2$$

Overall model (P_2) contains

$$S + 1 + (C-1)(C-2)\{2nm - 5(m+n) + 12\} + 4\{(n-3)(m-2) + (n-2)(m-3))\}$$

$$+ (C-1)(n-2)(m-2)\{(n-2)(m-2)-1\}/2$$

or $O(n^2m^2C)$ constraints.

Table 2.3 compares the total number of variables and constraints in the two models.

	model (P_1)	model (P_2)
X variables	$O(nmC)$	$O(nmC)$
Y variables		$O(n^2m^2C)$
constraints	$O(nmC^2)$	$O(n^2m^2C)$

Table 2.3: Approximate sizes of the two models

For example if we have a 13×13 grid that contains three types of species with three clusters, Table 2.4 shows the actual number of variables and constraints for models (P_1) and (P_2). Clearly, the second model contains a much larger number of variables and constraints. But if we consider the efficiency of solving these models, the second model is expected to be computationally preferable because it is a *linear* (mixed) integer programming model as opposed to a nonlinear, nonconvex integer model. This prediction is validated by the computational results presented in Chapter 3.

	model (P_1)	model (P_2)
X variables	507	507
Y variables		14961
constraints	613	16017

Table 2.4: Exact sizes of models (P_1) and (P_2) for a 13×13 grid

Chapter 3

Numerical Results

This chapter presents computational results with models (P_1) and (P_2) on a variety of data sets. Several software packages have been used to implement these models. Table 3.1 summarizes the different software components used to handle the interface, input data parameters, and output solutions, as well as the optimization solver. All experiments were carried out on a Dell Vostro 1400 computer with a Pentium-IV processor and 2 GB RAM.

Interface	Input data	Output solutions	Optimization solver
OPL	EXCEL	EXCEL	CPLEX

Table 3.1: Summary of the implementation characteristics

Each optimization problem was formulated using OPL (Optimization Programming Language), a modeling language for Linear and Integer Programming. OPL uses a solver called CPLEX to solve the mathematical models. OPL and CPLEX are produced by the ILOG Corporation (www.ilog.com) and are widely used for solving linear and integer problems.

We used CPLEX version 11.0 to solve our minimization models (P_1) and (P_2). CPLEX utilizes a variety of standard optimization routines, including branch and bound, to solve integer mathematical models. After finding a feasible solution, CPLEX either improves it or proves that it is optimal. To this end, at each step in the solution process, CPLEX tracks two numbers: the objective function value of the best known solution, and a certain lower bound on the optimal objective function value. The solution process terminates when the best known objective function value converges to the lower bound within a user-specified tolerance.

3.1 Comparison of models (P_1) and (P_2)

We first study models (P_1) and (P_2) on a sample study region with grid size 10×10. The study region is represented in Figure 3.1. In this example it is assumed that the reserve system contains three types of known species; $S1, S2$, and $S3$ denote species of type 1, type 2 and type 3 respectively. The species present in each site of the grid are indicated using $S1, S2$, and/or $S3$, while a zero indicates that no species are found in that particular site.

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	S1,S2,S3	0	0	S3	S1,S2,S3	S3	0	0	0
3	0	S1,S2,S3	S1,S2,S3	S3	S2,S3	S1,S3	0	S1,S2	S1	0
4	0	S1,S2,S3	S1,S2,S3	S1	S1	0	0	S1,S2,S3	S1,S2,S3	0
5	0	S1,S2,S3	S1,S2,S3	0	0	0	S1,S2,S3	S1,S2,S3	S1,S2,S3	0
6	0	0	0	S1,S2,S3	S1,S2,S3	S1,S2,S3	0	S1,S2,S3	S1,S2,S3	0
7	0	0	0	0	0	S2,S3	S2,S3	0	0	0
8	0	S1,S2,S3	S1,S2	S3	S1,S2,S3	0	S1,S2,S3	S3	S1,S2,S3	0
9	0	0	0	0	S2,S3	0	S3	S2	S3	0
10	0	0	0	0	0	0	0	0	0	0

Figure 3.1: 10×10 study region with 100 sites

Suppose that we need to conserve at least 10 sites for species of type 1, 8 for species of type 2 and 10 for species of type 3. As shown in Figure 3.1, initially the region contains 28 sites for species of type 1, 29 for species of type 2 and 34 for species of type 3. Also assume that the budgetary cost of conserving each site is 1. Thus our budget constraint simply becomes an upper bound on the number of selected sites. The maximum number of allowable clusters for this problem is set equal to 3.

Table 3.2 shows the total number of variables and constraints generated for the two models (P_1) and (P_2). We consider two different scenarios to illustrate this test problem and the results from the optimization models.

Case I

	model (P_1)	model (P_2)
X variables	300	300
Y variables		4257
constraints	328	4812

Table 3.2: Exact sizes of models (P_1) and (P_2) for a 10×10 study region

Here we suppose that the maximum number of selected sites is 10. The optimal solution for this case given by both models (P_1) and (P_2) is depicted in Figure 3.2. Since we allowed a maximum of three clusters ($C = 3$), the optimal solution contains two real clusters whose sites are denoted 2 and 3 respectively in Figure 3.2. Those sites of cluster 1 (the non-selected sites) are designated with a 1. In Figure 3.2, the two real clusters are shaded and exactly 10 sites are selected to satisfy the given species requirements. As mentioned the species coverage requirements for each type are 10, 8, 10 respectively and the selected two real clusters cover 10 of type 1, 10 of type 2, and 10 of type 3. Although we have a surplus for species of type 2, visually it is clear that the two selected real clusters display ideal compact shapes.

	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	2	2	1	1	1	1	1	1	1
4	1	2	2	1	1	1	1	3	3	1
5	1	2	2	1	1	1	1	3	3	1
6	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1

Figure 3.2: Optimal selection for Case I, 10×10 study region

Case II

Here a maximum of 15 sites can be selected and we use the same species coverage require-

ments. The optimal solution for this case given by both models (P_1) and (P_2) is depicted in Figure 3.3. Although 15 sites are allowed, the optimal solution produced used only 14 sites to satisfy all stipulated species coverage requirements. Here 10 species of type 1, 9 of type 2, and 10 of type 3 are covered by the optimal solution. Only a single real cluster is created, which is denoted 2 in Figure 3.3. In this case, although a single cluster is created, it displays a compact shape.

	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	2	2	2	2	1	1	1	1	1
4	1	2	2	2	2	1	1	1	1	1
5	1	2	2	2	2	1	1	1	1	1
6	1	1	1	2	2	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1

Figure 3.3: Optimal selection for Case II, 10×10 study region

As expected from Chapter 2, models (P_1) and (P_2) produce the same optimal solutions for both of the test cases. For Case I, the optimal solution value is displayed as 18000027.96. Here the first two nonzero digits correspond to the boundary length (18) of the selected clusters; after four following zeros the total distance (27.96) within the clusters is displayed. For Case II, the optimal solution value is 16000187.24; the first two nonzero digits correspond to the boundary length (16) of the selected cluster and after three following zeros the total distance (187.24) within the single cluster is displayed. Notice that the second optimal solution has a smaller boundary length than the first optimal solution. This shows that as we increase the number of allowable sites, we can improve the overall shape of the solution by creating a single compact reserve area. Since our model places an *upper bound* C on the number of clusters, this example shows that it may be beneficial to use fewer than the allowable number of clusters.

Model (P_1) took nearly 9 hours to provide the globally optimal solution whereas model (P_2) took only 4 seconds to provide the optimal solution with zero gap for Case I. For Case II model (P_1) took nearly 7 hours and 40 minutes while model (P_2) took 7 seconds. Despite the increased size of model (P_2) compared with model (P_1), model (P_2) took substantially less time to solve these test problems. After solving various test problems with different sizes using the two models, we found that solving model (P_2) was consistently and significantly faster than solving model (P_1). Thus we use model (P_2) in all our subsequent numerical investigations.

3.2 Varying the number of clusters

In this section we discuss how the optimal solutions behave when the allowable number of clusters C is changed. We illustrate this by considering a sample reserve system of size 13×13. We solve this test problem by specifying $C = 2,\ C = 3,\ C = 4$ and then $C = 5$. The study region is shown in Figure 3.4 and it contains three types of known species: $S1,\ S2,\ S3$. As before the species present in each site of the grid are indicated using $S1, S2, S3$ and a zero indicates that no species are found in that particular site. Suppose that we need to protect at least 50 sites for species of type 1, 52 for species of type 2 and 52 for species of type 3 by using at most 56 sites. The region initially contains 58 sites with species of type 1, 57 with type 2, and 57 with type 3. Here also it is assumed that the cost of conserving any site is 1.

Table 3.3 shows the total number of variables and constraints generated for this test problem by model (P_2) for $C = 2,\ C = 3,\ C = 4$, and $C = 5$. As we increase the number of clusters the optimal solution values given by model (P_2) and the CPU times taken (in seconds) are given in Table 3.4. The solution time appears to grow modestly with the size of the formulated model.

Number of clusters	Number of variables	Number of constraints
2	8039	8317
3	15468	16017
4	22897	24157
5	30326	32737

Table 3.3: Number of variables and number of constraints for the 13×13 grid when the number of clusters is changed.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	S2,S3	S1,S2,S3	S1,S3	S1,S2,S3	S1,S2,S3	0
3	0	0	0	0	0	0	0	S1,S2,S3	S1,S2,S3	S2,S3	S1,S2,S3	S1,S2,S3	0
4	0	S1,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	0	0	S1,S2,S3	S1,S2	S1,S2,S3	S2,S3	S1,S2,S3	0
5	0	S1,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	S2	0	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1	0	0
6	0	S1,S2,S3	S1,S2,S3	S1,S2,S3	S2,S3	0	0	S1,S2,S3	S1,S2,S3	S1,S2,S3	0	0	0
7	0	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1	S2,S3	S1	0	0	0	0	0
8	0	S1,S2,S3	S2	0	0	0	0	0	0	0	0	0	0
9	0	S1,S2,S3	S1,S2	0	S1	0	0	0	S1	S1	S3	0	0
10	0	0	0	S3	S3	S3	S1,S2	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	0	0
11	0	S2	S2	0	0	0	S2,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	0	0
12	0	S1	0	0	0	S1	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1,S2,S3	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.4: Sample conservation region with 169 sites

Number of clusters	Optimal solution value	CPU time (sec)
2	56008943.30	7
3	56003573.06	27
4	56001243.58	77
5	56001243.58	53

Table 3.4: Optimal solution values and CPU times for the 13×13 study region

Figure 3.5 shows the optimal solutions produced for this 13×13 test problem, as C is varied. Here, a 1 denotes those sites selected for Cluster 1, whereas 2-4 denote sites selected for Clusters 2-4, as applicable.

Figure 3.5(a) shows the clusters selected for conservation when $C = 2$ clusters are allowed. Out of 169 sites only 56 sites are needed to satisfy the species coverage and the budget constraints. Here all the selected sites are designated as belonging to Cluster 2. As we can see in Figure 3.5(a), Cluster 2 consists of three disconnected components. That is, model (P_2) as well as model (P_1) allows the possibility of disconnected clusters. This affects our interpretation of the objective function value 56008943.30 shown in the first row of Table 3.4. Here 56 is the overall boundary length of Cluster 2 and 8943.30 is the total distance within Cluster 2. Since Cluster 2 contains three components, the distance reported is measured between all distinct selected sites including those in different

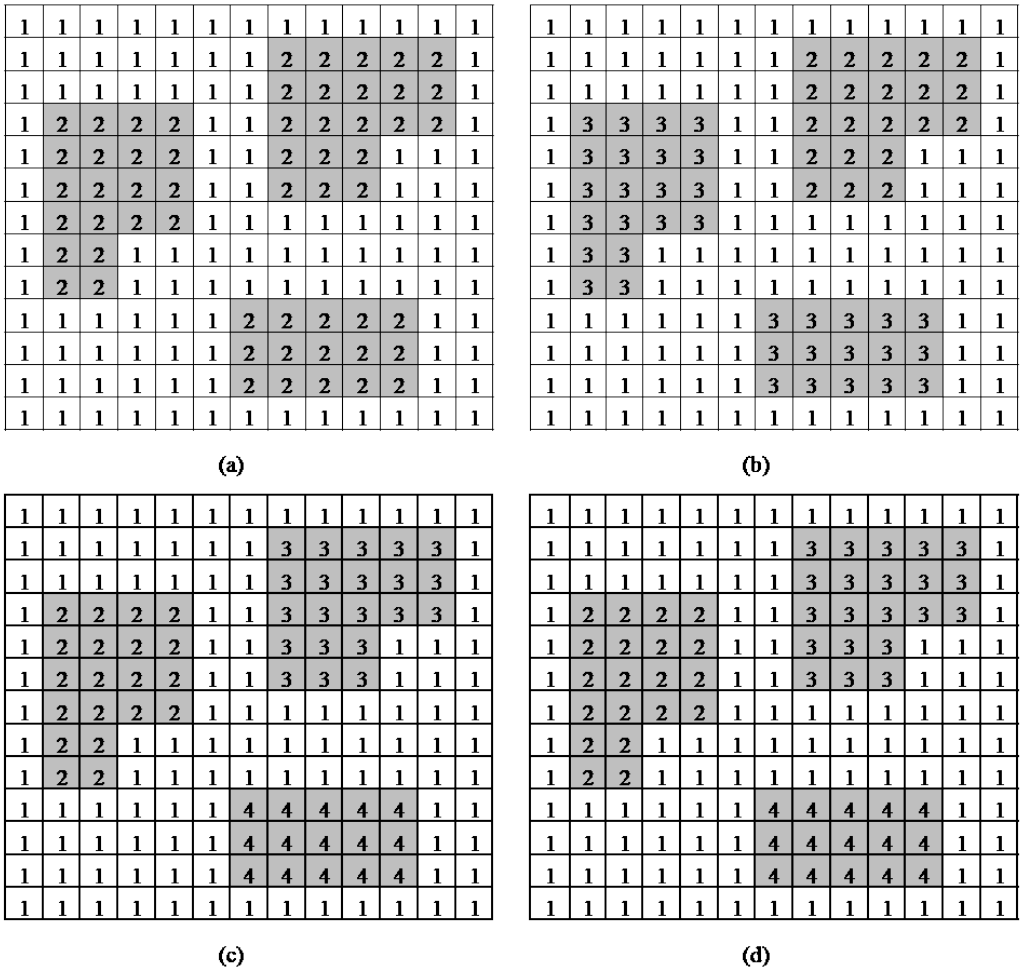

Figure 3.5: Clustered 13×13 sample study region

components. However the actual distance measured within each cluster is 1243.58.

Figure 3.5(b) shows the optimal solution when the number of allowable clusters is $C = 3$. As seen in the second row of Table 3.4, the optimal value of the objective function for Figure 3.5(b) is 56003573.06. Although we have the same clusters as before, in this case the total distance within clusters is reduced from 8943.30 to 3573.06. It should be noted that Cluster 2 has only one component whereas Cluster 3 has two components. That is, out of the 56 selected sites 21 sites are selected for Cluster 2 and 35 sites are selected for Cluster 3. However the actual distance measured within each cluster remains 1243.58.

Figure 3.5(c) shows the optimal solution when the number of allowable clusters is $C = 4$. Here this solution identifies three completely connected real clusters. The third row of Table 3.4 shows the optimal solution value 56001243.58. In this case we have same clusters as before but the total distance within clusters is reduced from 8943.30 to 1243.58. It should be noted that here 20 sites are selected for Cluster 2, 21 sites are selected for Cluster 3 and 15 sites are selected for Cluster 4.

Finally consider Figure 3.5(d) which shows the optimal solution when the number of allowable clusters is $C = 5$. Figure 3.5(d) shows exactly the same real clusters as Figure 3.5(c) and the same optimal objective value. This indicates that the optimal solution to this test problem could not be improved by increasing the value C.

The changes in the reported optimal objective value with C are depicted in Figure 3.6 for this 13×13 test problem. As shown in this graph, the optimal objective value is a nonincreasing function of the number of clusters. It is also important to note that when $C = 2$ the optimization model (P_2) produces a reasonable set of clusters — in this case optimal for $C = 4$. Therefore in general, even for small values of C, the model can identify a useful clustering of reserve sites. Moreover, as seen in Table 3.4, it can be computationally advantageous to use $C = 2$ in our model.

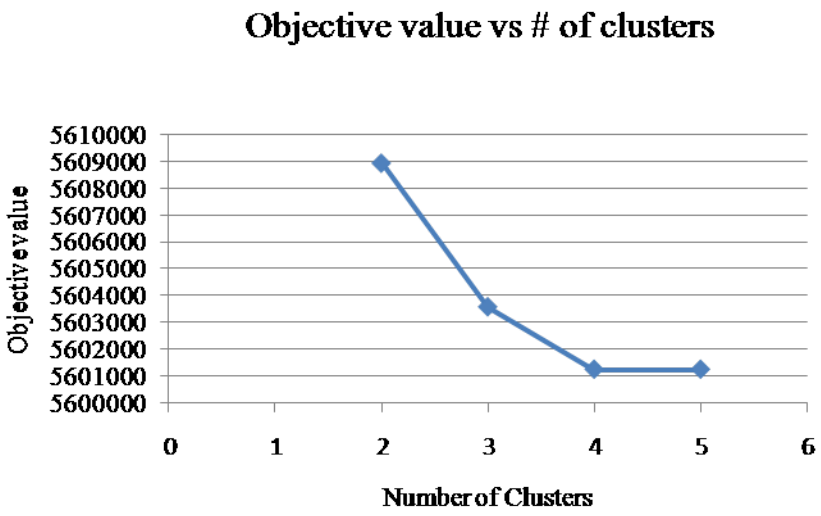

Figure 3.6: Graphical representation of optimal objective values for the 13×13 test problem

It is also possible to obtain different clusters as the value C increases. We illustrate this by

considering a sample reserve system of size 16×16, shown in Figure 3.7. We solve this test problem by specifying $C = 2,\ C = 3$ and $C = 4$. As we increase the number of allowable clusters for the 16×16 grid, the optimal solution values given by model (P_2) and the CPU times taken (in seconds) are given in Table 3.5. Again we notice that the CPU times increase only modestly with problem size.

	1	2	3	4	5	6	7	8	9	10	11	12	15	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	S1	S2,S3	S1,S2,S3	S1	S1,S3	S1,S2,S3	S3	S1,S2	0	S1,S3	0	S3	0	S2	0
3	0	S1	S1,S2	S2,S3	S1,S2,S3	S1,S3	S2,S3	S2,S3	S3	S1	0	S1,S3	S1	S2	S2,S3	0
4	0	S1,S2,S3	S2,S3	S1,S2,S3	S1,S2,S3	S3	S2,S3	S3	S1,S2	S1,S2	S1,S2,S3	S1,S2	S2,S3	S1	S1	0
5	0	S1,S2	0	S3	S1	S1,S2,S3	S1	0	S2	S2	0	S3	S1,S3	0	S3	0
6	0	S1,S2,S3	S2,S3	S3	0	0	S2	0	0	S1,S2	S2,S3	S2	S1,S2,S3	S1	0	0
7	0	0	S1,S2,S3	S1,S2	S2	S3	S3	S1	S1	S2	S2	S1,S2,S3	S1,S2	S2	S2,S3	0
8	0	0	S1,S3	S1,S3	0	S3	0	S2	0	0	0	S1	S1,S2,S3	S1	S2	0
9	0	S1,S2,S3	S1,S2,S3	S1	S3	S2	S1	0	S3	0	S3	S2	S2	S3	0	0
10	0	S2,S3	S2	S1	S1,S3	S2	0	S2,S3	0	S1	S2,S3	S1	S1,S3	S3	S1	0
11	0	S3	S3	0	0	S2,S3	S1,S2	S1	S2	S1,S2,S3	S2	S1,S3	S1	0	0	0
12	0	S2	0	S1	S1,S3	S2	S1,S2,S3	S1,S2,S3	S1,S3	S1,S2	S1,S2,S3	S2	S3	S1	S1	0
13	0	0	0	S1,S3	S1,S3	S2	S1,S3	S1,S2	S1,S2,S3	S2,S3	S1,S2,S3	S1,S3	S1	S3	S3	0
14	0	S2	S3	0	S1,S3	S1,S3	S1	S1,S2,S3	S1,S2,S3	S1,S2,S3	S1	S1,S2,S3	0	0	S2	0
15	0	S1	0	S1,S2	S3	S2	0	0	S1	S1,S2	S1	S2	S2,S3	0	S1	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.7: Sample conservation region with 256 sites

Number of clusters	Optimal solution value	CPU time (sec)
2	56013935.90	16
3	56003483.86	77
4	56003483.86	108

Table 3.5: Optimal solution values and CPU times for the 16×16 study region

Figure 3.8(a)–(c) shows the clusters selected for conservation when $C = 2,\ C = 3$ and $C = 4$ respectively. Notice that slightly different clusters are produced for $C = 2$ and $C = 3$. Namely one site has migrated between the components of Figure 3.8(a) to yield Figure 3.8(b). That is, site $(9, 4)$ of Cluster 2 in Figure 3.8(a) has migrated to position $(15, 11)$ of Cluster 3 in Figure 3.8(b). It is

important to notice, however, that the boundary length (56) remains unchanged, while the reported distance value is reduced from 9935.90 to 3483.86; see Table 3.5. When $C = 2$, the actual distance measured within the two identified components of Cluster 2 is 3509.50, which is slightly larger than the value 3483.86 for $C = 3$ and $C = 4$. In this example, increasing the number of clusters did decrease the total distance within clusters. However the total boundary length remains the same.

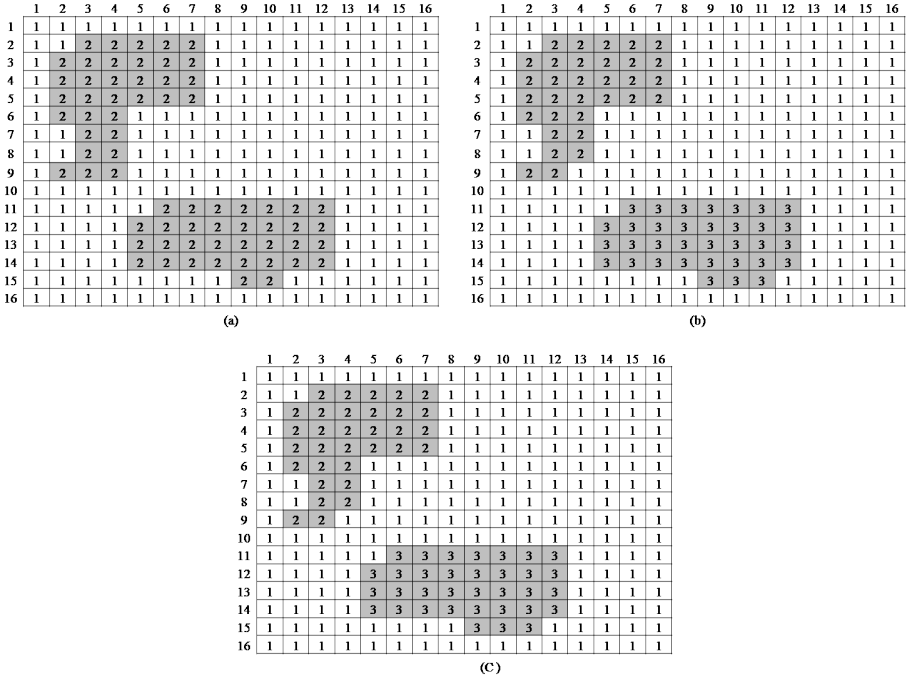

Figure 3.8: Clustered 16×16 sample study region

These examples suggest that as we increase the number of clusters, the boundary length of the selected real clusters remains the same. This behavior in general holds.

We first show that the boundary length stays the same as C is varied. Notice that an optimal solution $\boldsymbol{x}^{(2)}$ for $C = 2$ is always a feasible solution for $C > 2$. Since the objective function of model (P_2) has minimization of the boundary length as the primary criterion, the boundary length for $\boldsymbol{x}^{(2)}$ is at least as large as the boundary length of an optimal solution $\boldsymbol{x}^{(C)}$ for $C > 2$. But the optimal solution $\boldsymbol{x}^{(C)}$ for $C > 2$ is also a feasible solution for $C = 2$: just set all of the selected sites to belong to Cluster 2. Therefore the boundary length for $\boldsymbol{x}^{(C)}$ is at least as large as the boundary length of $\boldsymbol{x}^{(2)}$. This implies that optimal solutions for $C = 2$ and $C > 2$ must have the same boundary length.

Now suppose that the optimal solution $\boldsymbol{x}^{(2)}$ for $C = 2$ consists of $(k-1)$ disjoint components

(all in Cluster 2). Consider an optimal solution $\boldsymbol{x}^{(k)}$ for $C = k$. We already know that $\boldsymbol{x}^{(2)}$ and $\boldsymbol{x}^{(k)}$ have the same boundary length. Since $\boldsymbol{x}^{(2)}$ provides a feasible solution for $C = k$, and since the objective function in model (P_2) for $C = k$ measures the actual within cluster distances for $\boldsymbol{x}^{(2)}$, we see that $\boldsymbol{x}^{(2)}$ has a within cluster distance at least as large as that for an optimal solution $\boldsymbol{x}^{(k)}$.

In other words we see that using $C = 2$ provides a clustering with the optimal boundary length, but it need not produce the smallest within cluster distance possible for a larger value of k. (This can occur for example when the components of Cluster 2 are forced to be close to one another in order to minimize the objective function—based on all sites in Cluster 2 and not simply sites within components of Cluster 2.)

Since model (P_2) produces optimal solutions with the same boundary length for all $C \geq 2$ and since the optimal solutions when $C = 2$ provide reasonable clusters, we prefer solving our models for $C = 2$ instead of using higher values of C. This will enable us to reduce the required CPU time significantly.

3.3 Varying the coverage of species

In this section we discuss how the optimal solution behaves when the number of species of each type is changed. We illustrate this by considering the data set presented in [16]. In this example 16 hypothetical species (A through P) are distributed across a reserve system containing 100 sites. The region is shown in Figure 3.9. Since our formulation assumes a border of non-selected sites, we therefore treat this test problem as a 12×12 grid. We now consider two scenarios.

Cover each species at least once

First suppose we need to cover at least one of each species. The optimal solution for this case when $C = 2$ is depicted in Figure 3.10. Model (P_2) produces a solution with three clusters, which is the same solution as that reported in [16]. Also, it should be mentioned that increasing the maximum number of allowable sites from 3 to 30 does not change the optimal solution.

Cover each species at least twice

Now suppose we need to cover at least two of each species. For this case we obtain three different solutions as the maximum number of allowable sites is changed for $C = 2$. These optimal

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	KOP	AIJLN	HMOP	HIP	IJMNP	DKNOP	ABILMP	IKLMOP	NOP	CKNP	0
3	0	LMN	GHOP	ABOP	NOP	AELMNP	CKNOP	AFMO	EIL	MNP	AJKLMNOP	0
4	0	KMO	BNO	FKNOP	O	IKNOP	OP	GO	DJKNP	MOP	LNO	0
5	0	P	FKOP	NOP	FIP	BJKO	IKNO	ILOP	MOP	DEFGJOP	O	0
6	0	DNP	EO	JMNOP	IJMNP	MNOP	ACGHKNP	KMOP	CMOP	BKP	M	0
7	0	BNOP	MOP	CDGHKMOP	ABCGLMP	DNO	EMOP	OP	NOP	NOP	DHMNO	0
8	0	JKLMOP	LMNO	EHMO	ACEGMO	NOP	JNOP	DNP	JMNOP	BKLNP	MOP	0
9	0	IKLNP	NP	DOP	HN	DMNP	AGHLOP	LP	IOP	NOP	MNOP	0
10	0	KMNOP	GHIJMOP	NOP	AFKMNP	LOP	ACEGOP	IMNP	HMOP	DHLMOP	ADIJKLNOP	0
11	0	IMNOP	GMNO	MOP	KMNOP	MOP	DNO	MNOP	KNOP	DLMOP	IMN	0
12	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.9: Sample reserve system from [16]

solutions are depicted in Figure 3.11. When the number of allowable sites is 6, Figure 3.11(a) shows the optimal solution and when the number of allowable sites is 7 or 8, Figure 3.11(b) shows the optimal solution. Figure 3.11(c) shows the optimal solution when the number of allowable sites is more than 8. As we increase the number of allowable sites, the clusters become fewer in number and more compact in shape.

3.4 More extensive computational results

To test the computational performance of model (P_2), we created hypothetical sample grids of various sizes. Three clusters with arbitrarily predefined shapes were created. We generated problems having three species. For each species type we used a random number generator to assign random numbers between zero and one for each grid site. For sites within a predefined cluster, we randomly choose the species to occupy that site with a specific probability p: that is, if the random number generated for that site was less than p. For sites not within a predefined cluster, we choose

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	2	1	1
6	1	1	1	1	1	1	2	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3.10: Optimal clustered regions when $C = 2$ for the data set from [16] with each species covered at least once

the species to occupy that site with a specific probability q where $q < p$.

Here we studied two different grids with sizes 16×16 and 18×18. For practical reasons the run time was limited to 1 hour and at most 36 selected sites were allowed in these problems. The number of variables and the number of constraints appearing in model (P_2) are given in Table 3.6 when $C = 2$.

	model 16×16	model 18×18
Number of variables	20351	20830
Number of constraints	24249	34892

Table 3.6: Exact sizes of 16×16 and 18×18 models when $C = 2$

Also, three predefined clusters were created having boundary length 48 and total within cluster distance 387.28. Figure 3.12 displays for the 16×16 grid the predefined clusters, which are shown as shaded.

Table 3.7 displays the findings of 10 experimental runs for the 16×16 grid and the 18×18 grid when $C = 2$. As shown in Table 3.7, the 16×16 problems and the 18×18 problems could be solved exactly within the given time requirement. Throughout we set $p = 0.7$ and allowed q to vary as shown in the first column of Table 3.7. The second column shows the required CPU times in seconds. The last two columns show the optimum boundary length BL and the distance DIS within selected clusters.

The optimal solutions for the 16×16 and 18×18 test problems when q varies are depicted

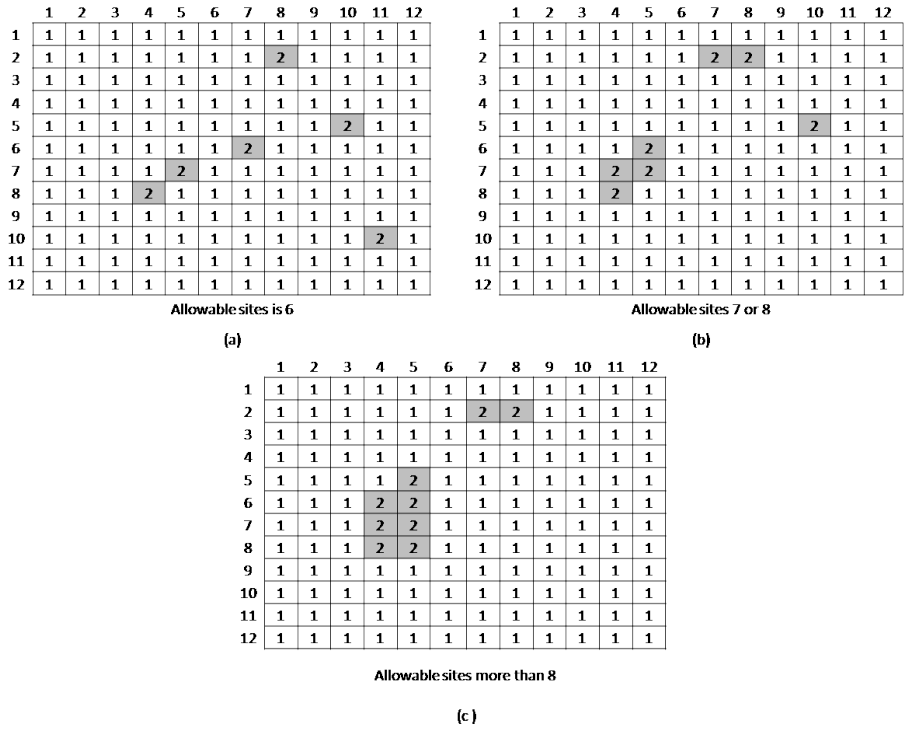

Figure 3.11: Optimal clustered regions when $C = 2$ for the data set from [16] with each species covered at least twice

in Figure 3.13 and Figure 3.14 respectively. In these figures, the shaded cells indicate those sites selected for the real clusters. Note that as q increases, the number of species outside the predefined clusters in Figure 3.12 will generally increase. As a result some clusters merge together and this is verified in Figure 3.13. For example Figure 3.13(c) and Figure 3.13(d) have two real clusters while Figure 3.13(e) has only one real cluster. Also Table 3.7 shows that for all these cases the optimal boundary lengths are smaller than the predefined boundary length 48.

We use the same predefined clusters for the 18×18 test problems except that a border of non-selected cells is placed around the 16×16 grid. Here also we notice that clusters can merge together for higher values of q. Also Table 3.7 shows that for all these cases the optimal boundary lengths are smaller than the predefined boundary length 48.

In all cases, the clusters obtained tend to be fairly compact and reasonable (even though we only specified $C = 2$ in solving our models). In summary, the computational results for the 16×16 and 18×18 grids show that even with $C = 2$ we can obtain reasonable clusters in an

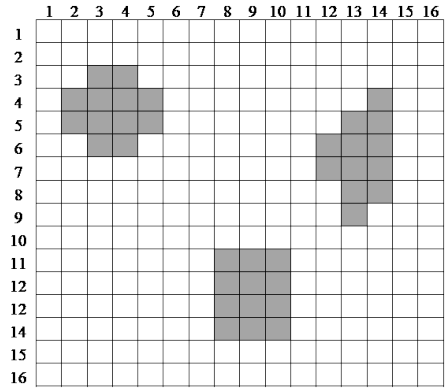

Figure 3.12: Predefined clusters for a 16×16 grid

q	CPU time	Optimal BL	Optimal DIS
16×16 with zero gap			
0.20	72	46	364.73
0.25	81	46	387.74
0.30	190	38	751.39
0.35	248	40	794.99
0.40	298	34	2649.80
18×18 with zero gap			
0.20	141	46	368.58
0.25	1140	40	881.49
0.30	1041	44	392.46
0.35	241	44	680.08
0.40	480	36	740.38

Table 3.7: Computational results for 16×16 and 18×18 grids

acceptable amount of computational time. These results indicate that model (P_2) can reproduce compact solutions that tend to conform to our predefined clusters, given the statistical uncertainty associated with the generation method.

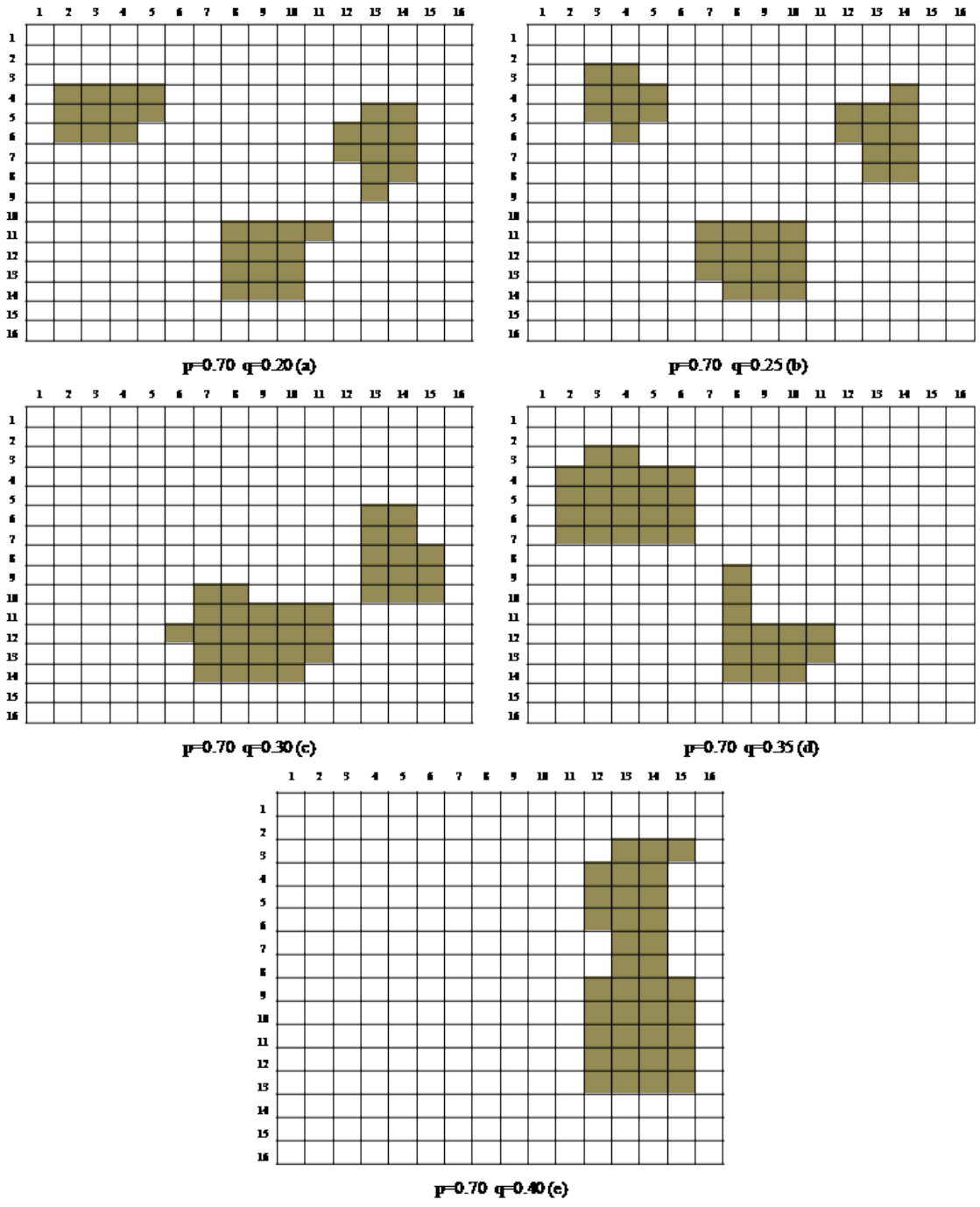

Figure 3.13: Optimal clusters for the 16×16 grid when $C = 2$ and q varies

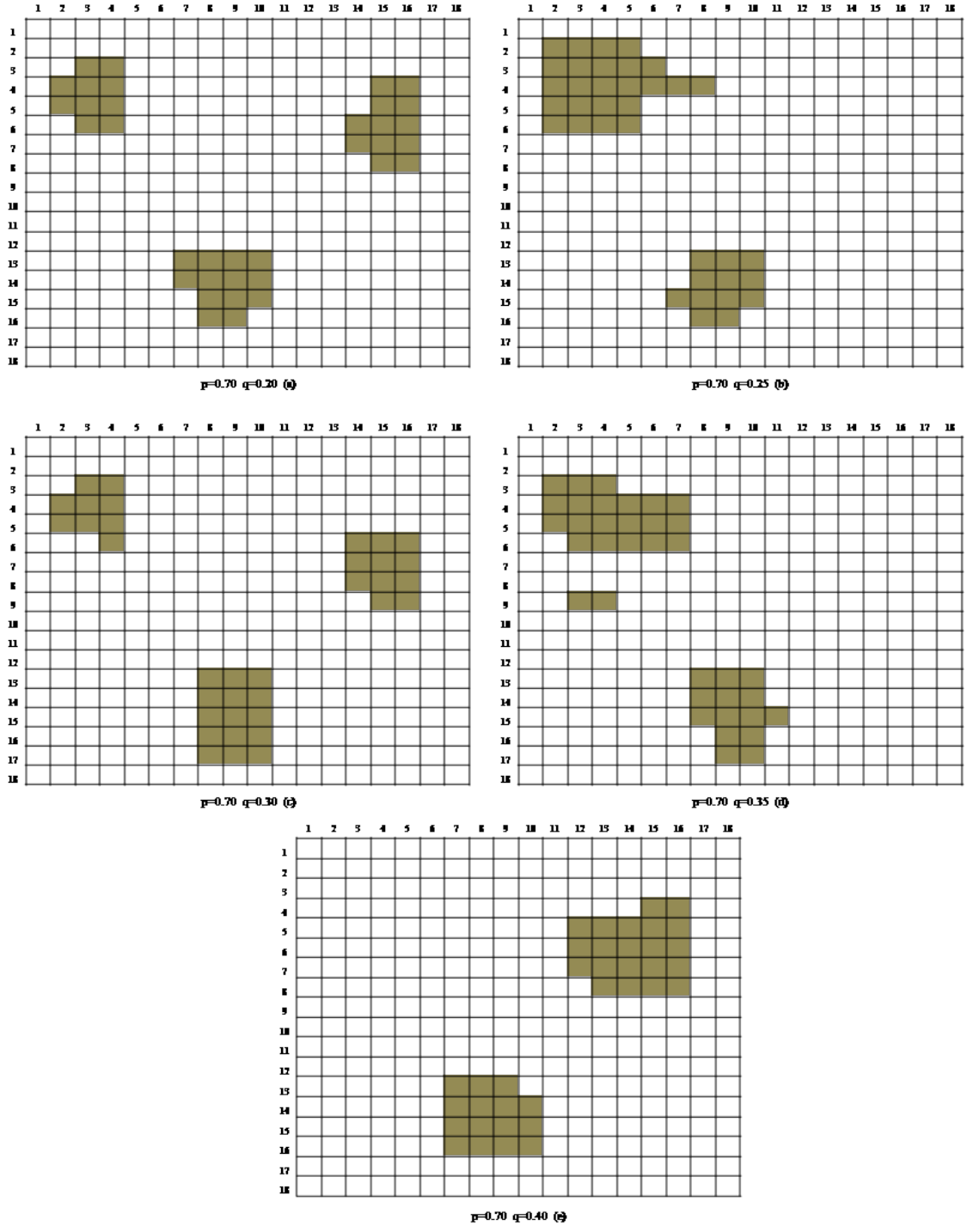

Figure 3.14: Optimal clusters for the 18×18 grid when $C = 2$ and q varies

Chapter 4

A Heuristic Solution Method

Two general types of approaches have been devised to solve reserve design problems: exact algorithms and heuristic (non-exact) algorithms. Although exact algorithms can identify an optimal solution, for large reserve design problems it is difficult (and often impossible) to find an optimal solution in a reasonable amount of time. Heuristics, on the other hand, can provide a number of good, near-optimal solutions, and can these also be generated very quickly. As a result heuristics are generally preferred over exact algorithms for realistic-sized problems.

Several heuristics have been developed for reserve design problems [2, 8]. Here we develop a heuristic solution method by using three procedures based on the proposed model (P_2). In the following sections we explain these three procedures.

4.1 Linear programming relaxation – Procedure 1

Relaxation problems give us useful information in an attempt to solve integer programming problems. We can directly obtain lower bounds for minimization integer programming problems by examining their associated relaxations. Specifically, the linear programming relaxation of model (P_2) is obtained by replacing the binary constraints $X_{cij} \in \{0, 1\}$ with $0 \leq X_{cij} \leq 1$. That is, we relax the integer variables X_{cij} to continuous variables. Optimization software typically needs much less time to find optimal solutions to relaxed problems, which are then linear programming, as opposed to integer programming problems. To illustrate, the third and fifth columns of Table 4.1 show the CPU times (in seconds) to solve the relaxed model for (P_2) and the original model

(P_2), using the test problems discussed in Section 3.4 (with $C = 2$). Table 4.1 clearly shows that the relaxed problems require substantially less solution time than the integer problems. The second and fourth columns of Table 4.1 show the optimal solution values for the relaxed problems and the integer problems respectively. As expected the optimal solution values for the relaxed problems provide lower bounds on the optimal solution values for the corresponding integer problems.

q	Optimal relaxed solution value	CPU time (sec) for relaxed problem	Optimal integer solution value	CPU time (sec) for integer problem
16×16 grids				
0.20	45004188.58	0.12	46003966.42	72
0.25	43696085.49	0.07	46003936.92	81
0.30	35646686.40	0.11	38002697.62	190
0.35	37158013.12	0.09	40003516.93	248
0.40	31095384.02	0.11	34002649.78	298
18×18 grids				
0.20	44576642.79	0.12	46004952.15	141
0.25	36404423.83	0.15	40004231.15	1140
0.30	39845192.51	0.11	44005443.91	1041
0.35	40719408.07	0.13	44004287.26	241
0.40	34719634.25	0.21	36004012.11	480

Table 4.1: Comparison of CPU times and optimal solution values for 16×16 and 18×18 grids: relaxed and integer problems

Although the relaxed optimal objective value provides a valid lower bound on the integer optimal solution value, the relaxed problems do not in general provide feasible integer solutions to the original integer mathematical programming problems. For example Figure 4.1 shows the optimal integer solutions and optimal relaxed solutions for two 10×10 test problems. Here a 1 denotes those sites selected for Cluster 1 and a 2 denotes those sites selected for Cluster 2. All fractional values shown are the non-integral values of X_{cij} in the optimal solution of the relaxed model.

Figure 4.1(a) and Figure 4.1(b) show the integer optimal solution and the relaxed optimal solution for the first 10×10 test problem, while Figure 4.1(c) and Figure 4.1(d) show the integer optimal solution and the relaxed optimal solution for the second 10×10 test problem. The solutions in Figure 4.1(a)–(b) are in fact the same. Thus in this case the optimal relaxed solution is equal to the optimal integer solution. However the solution shown in Figure 4.1(d) is fractional and thus does not provide a feasible solution to the original model. On the other hand, the sites selected for Cluster 2 in the relaxed model provide a first-order approximation to the optimal Cluster 2 shown

in Figure 4.1(c).

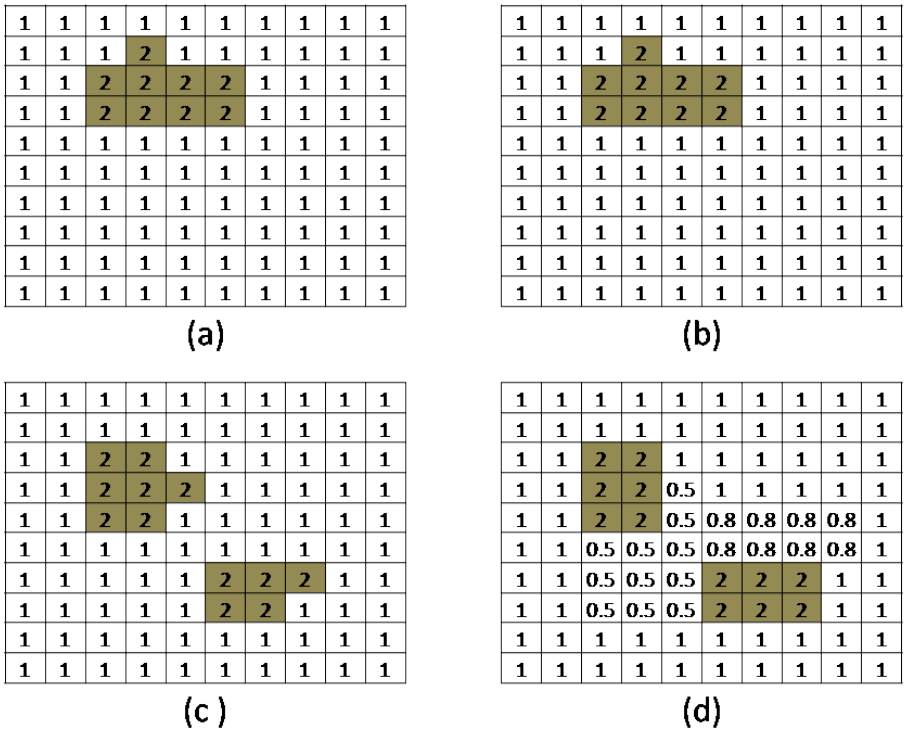

Figure 4.1: Integer solutions (a), (c) and relaxed solutions (b), (d) for a 10×10 grid

Additional testing was conducted on 20 instances of 10×10 grids, all with same predefined clusters, $p = 0.7$, and $q = 0.3$. We found that 16 of the 20 optimal solutions for the relaxed model were in fact integer, a fairly surprising result. More generally, however, relaxed problems do not produce feasible (integer) solutions to model (P_2), but often the optimal solution to the relaxed problem has a number of X_{cij} variables equal to 1. Thus we can use those sites (i, j) with $X_{cij} = 1$ $(c > 1)$ in the relaxed optimal solution as a starting point to obtain a feasible solution to the integer problem. Using the sites so identified in the optimal solution to the relaxed problem, we have developed a procedure to obtain a good *feasible* solution to the given integer problem (P_2). This procedure is discussed in the following section.

4.2 Obtaining a feasible solution – Procedure 2

Here we start with an infeasible solution given by the relaxed model. Namely, we begin with sites (i, j) having $X_{cij} = 1$ $(c > 1)$ in the relaxed optimal solution. Since the solution is not feasible, there is at least one species for which the required number of covered sites is not achieved. For example suppose that we need to conserve at least 5 sites for species of type 1, 8 sites for species of type 2 and 10 sites for species of type 3. If a relaxed solution restricted to sites with $X_{cij} = 1$ $(c > 1)$ covers only 2 sites for species of type 1, 5 sites for species of type 2 and 8 sites for species of type 3, then we need to cover at least 3 sites for species of type 1, 3 sites for species of type 2 and 2 sites for species of type 3 to obtain a feasible solution to model (P_2). Thus the *residual value* for each species is 3, 3, 2 respectively, giving the residual vector $r = [3,\ 3,\ 2]$. In order to reduce the residual values, we start with sites (i, j) having $X_{cij} = 1$ that were selected by the relaxed solution. We then consider all non-selected sites that are adjacent existing clusters as possible candidates to obtain a feasible solution. These candidates are prioritized based on the following factors, which will be explained further in subsequent sections:

(1) Mean and variance of the residual vector

(2) Boundary length

(3) Fractional value of the relaxed solution

(4) Distance

4.2.1 Mean and variance of the residual vector

Our first priority is to obtain a feasible solution. When we add to an existing cluster a non-selected site that is adjacent to a site in the current solution, we first consider the additional coverage of each species by the non-selected site. If we have several sites which cover the same maximum number of additional species, then to select a site among them, we consider the following steps. For each site v determine the additional species that it covers. Namely, let $\overline{a}_v$ be the vector giving the additional sites covered for the various species. Then we select a site v with the most additional coverage in order to minimize $\overline{r}_v = \overline{r} - \overline{a}_v$ where r is the current residual vector. In the case of ties, we select a site with minimum residual variance. Recall that for a data set with values $b_1, \ldots, b_n$, the variance is given by $\sum (b_i)^2/n - \overline{b}^2$ where $\overline{b}$ denotes the mean of the data set and n is

the number of data values. Since in the case of ties $\overline{r}_v$ is the same for several sites v, minimizing the variance is equivalent to minimizing the sum of squared components of $\overline{r}_v$. For example, consider a reserve that contains three types of known species and suppose that the current solution creates the residual vector $\overline{r} = [5,\ 4,\ 5]$. In this vector the first, second and third positions show the residuals for species of type 1, type 2 and type 3 respectively. Now suppose that we have two non-selected sites v_1 and v_2 which cover two additional species: namely $\overline{a}_{v_1} = [1,\ 0,\ 1]$ and $\overline{a}_{v_2} = [0,\ 1,\ 1]$ respectively. If we select the site v_1, the new residual vector is $\overline{r}_{v_1} = [4,\ 4,\ 4]$ and the sum of squared components of $\overline{r}_{v_1}$ is $4^2 + 4^2 + 4^2 = 48$. If we select the site v_2, the new residual vector is $\overline{r}_{v_2} = [5,\ 3,\ 4]$ and the sum of squared components of $\overline{r}_{v_2}$ is $5^2 + 3^2 + 4^2 = 50$. In this case we choose the site v_1 since it has the minimum sum of squared values compared with v_2.

4.2.2 Boundary length

We have developed our mathematical models for selecting reserve sites so that the selected sites are clustered into compact groups. As explained in Chapter 1 the boundary length of the selected clusters is of primary importance. Thus we consider as the next most important factor the boundary length, following the critical need to achieve feasibility (embodied in reducing the residual vector). This means that if several sites adjacent to the current clusters all produce minimum residual mean and variance, we select one that adds the minimum boundary length to its adjacent cluster.

4.2.3 Fractional value of the relaxed solution

We develop our heuristic based on the linear relaxation solution. Although there is no definitive theoretical argument, we believe that the linear relaxed value of a site is directly related to the usefulness of the site in creating a good solution to (P_2). For example consider a variable X_{cij} which has a fractional value in the linear relaxed solution and suppose the coefficient of the variable X_{cij} in the objective function is α. If we increase the variable X_{cij} from its current value to 1 (i.e., select site (i, j) for Cluster c), the objective function value will increase by approximately $(1 - X_{cij}) * \alpha$. Thus a fractional value of X_{cij} close to one will minimize this increase as it produces a solution with decreased infeasibility. Therefore if several adjacent sites provide the same minimal values for the previous two factors, we select one that has the maximum fractional value in the

relaxed solution.

4.2.4 Distance

In the case that all the above three prioritized criteria are satisfied by several sites, we consider the effect of potential sites on the overall distance measure DIS. As mentioned in Chapter 1, the distance between sites in a cluster is also important in defining compact clusters. Thus if all the above three prioritized criteria are satisfied by several sites, we consider how much adding a new site will increase the overall distance measure DIS. Then among the current candidate sites we pick one that gives the minimum additional distance within its adjacent cluster.

Choosing at each step a site according to these four prioritized criteria provides a reasonable method to obtain a feasible solution. The structure of this procedure is summarized by the diagram in Figure 4.2.

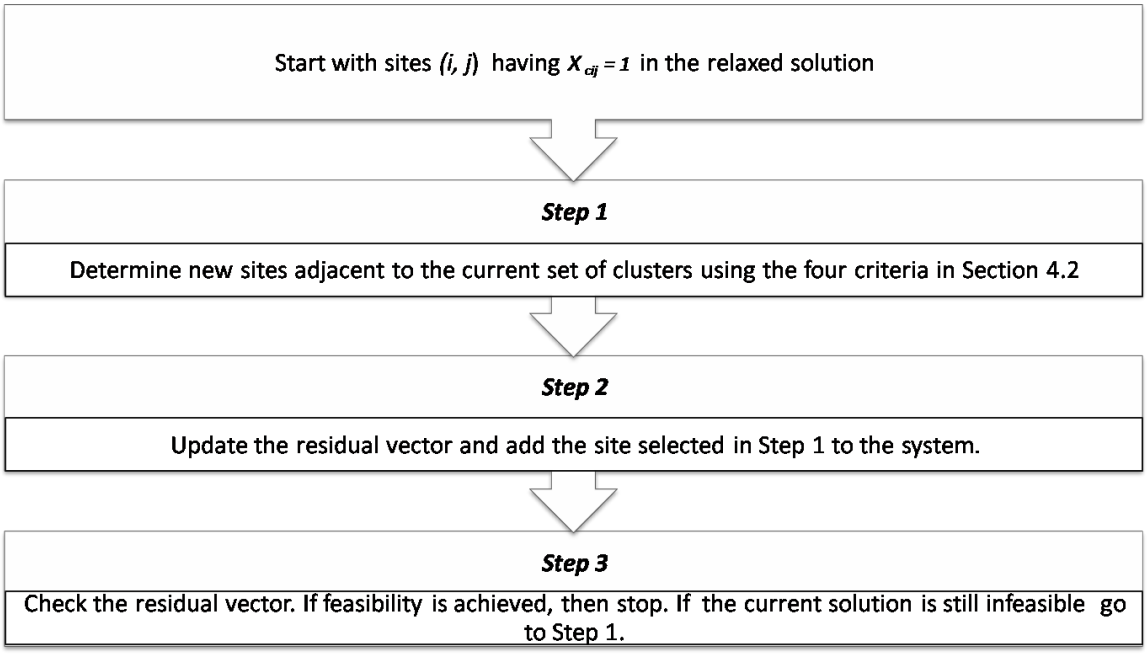

Figure 4.2: Heuristic flow diagram

4.3 Improving a feasible solution – Procedure 3

Interchange heuristics are commonly used in metaheuristic procedures for finding improved solutions to discrete optimization problems. In general, interchange heuristics for reserve site selection problems begin with a feasible solution. Here we can use the feasible solution produced by Procedure 2. Then we define a set of *feasible moves*, so that we can examine other nearby solutions.

To this end define the set $SE = \{u$: u is a selected site on the boundary of an existing cluster$\}$ and the set $NS = \{v : v$ is a non-selected site adjacent to some site $u \in SE\}$. Here we focus on interchange heuristics that swap one site currently in the set SE with another site in the set NS. Consequently a feasible move m_{uv} removes site u from an existing cluster and adds site v to its adjacent cluster if this modification maintains feasibility. We improve the solution by considering two criteria. First we consider the reduction in boundary length due to such an interchange. If several interchanges yield the same reduction in boundary length then we consider the reduction in total distance by the interchange. We first discuss the reduction in boundary length, using the algorithm below. This algorithm produces an improvement network H that summarizes all feasible moves relative to the current system.

Improvement Algorithm

For all sites $u \in SE$ and $v \in NS$:

> **Step 1**: Check whether the move m_{uv} is feasible. If so calculate the reduction ΔBL_{uv} in boundary length for the updated system in which the site u is removed from the system and the site v is added to its adjacent cluster.
>
> **Step 2**: If the feasible move m_{uv} has $\Delta BL_{uv} \geq 0$, add the edge (u, v) with weight ΔBL_{uv} to the improvement network H. Also calculate the the change in total distance ΔDIS_{uv}.

To obtain a improved solution based on interchanges we find a maximum weight bipartite matching [1] in the improvement network H.

Maximum Weight Bipartite Matching

We first introduce some graph-theoretic terminology. A graph H is *bipartite* if its vertex

set can be partitioned into two sets U and V such that no edge of H has both end points in the same set of the partition. The value on an edge is called the weight of the edge. A *matching* M is a collection of edges in H such that every vertex of H is incident to at most one edge of M. A *maximum weight matching* is a matching having the largest total weight. The complexity of an algorithm [1] for finding a maximum bipartite matching is $O(NM + N^2 \log N)$ where N is the order of the node set and M is the order of the edge set.

4.4 Numerical example for the heuristic algorithm

To provide an illustrative numerical example, consider the second study region used in Section 3.2. For that 16×16 region the relaxed solution for model (P_2) is given in Figure 4.3. The optimal relaxed solution when $C = 2$ contains two disconnected components containing sites (i, j) with $X_{2ij} = 1$. Those sites are shaded in Figure 4.3. The given species coverage requirements for this test problem are to conserve at least 49 sites for species of type 1, 41 sites for species of type 2 and 44 sites for species of type 3. But the relaxed solution with $X_{2ij} = 1$ covers only 44 sites for species of type 1, 37 sites for species of type 2 and 39 sites for species of type 3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
3	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
4	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
5	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
6	0	1	1	1	0	0	0	0	0	0	0	0.21	0.21	0	0	0
7	0	0	1	1	0	0	0	0	0	0	0	0.21	0.21	0	0	0
8	0	0	1	1	0	0	0	0	0	0	0	0.21	0.21	0	0	0
9	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0.87	1	1	1	1	1	1	0	0	0	0
12	0	0	0	0	0.87	0.87	1	1	1	1	1	1	0	0	0	0
13	0	0	0	0	0.87	0.87	1	1	1	1	1	1	0	0	0	0
14	0	0	0	0	0.87	0.87	1	1	1	1	1	1	0	0	0	0
15	0	0	0	0	0	0	0	0	0.15	0.15	0.15	0.15	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.3: Relaxed linear solution for the 16×16 grid in Section 3.2

To obtain a feasible solution we still need to cover at least 5 sites for species of type 1, 4 sites for species of type 2, and 5 sites for species of type 3. Thus for this relaxed solution defined by $X_{2ij} = 1$, the residual vector is $[5, 4, 5]$ and therefore the relaxed solution is not feasible for model (P_2). Therefore starting with the sites having value 1 (Procedure 1) we apply the heuristic

Procedure 2 to obtain a feasible solution.

The steps of Procedure 2 are explained using Figure 4.4(a)–(h). The first, second, third, and fourth columns of Table 4.2 show the succession of new sites added, the additional coverage provided by the new site, the updated residual vectors, and the updated system respectively. Each new site is selected according to the prioritized procedure explained in Section 4.2.

For example initially the site $(14, 6)$ which is adjacent to $(14, 7)$ is selected and its coverage is [1, 0, 1]. That is, it covers 1 site for species of type 1 and 1 site for species of type 3. The updated residual vector is [4, 4, 4] and the updated system is shown in Figure 4.4(a). Then based on the updated system in Figure 4.4(a), the site $(11, 6)$ which is adjacent to $(11, 7)$ is selected and its coverage is [0, 1, 1]. The updated residual vector and the updated system are shown in the second row of Table 4.2 and in Figure 4.4(b) respectively. In a similar manner we end up adding 8 new sites to the relaxed solution, then achieving feasibility: the current residual vector is $r = [0, 0, 0]$. This feasible solution, shown in Figure 4.4(h), has boundary length 66 and within cluster distance 3559.42.

New site	Additional coverage of the new site	New residual	Updated system
$(14, 6)$	[1, 0, 1]	[4, 4, 4]	Figure 4.4(a)
$(11, 6)$	[0, 1, 1]	[4, 3, 3]	Figure 4.4(b)
$(14, 5)$	[1, 0, 1]	[3, 3, 2]	Figure 4.4(c)
$(15, 10)$	[1, 1, 0]	[2, 2, 2]	Figure 4.4(d)
$(13, 5)$	[1, 0, 1]	[1, 2, 1]	Figure 4.4(e)
$(10, 8)$	[0, 1, 1]	[1, 1, 0]	Figure 4.4(f)
$(12, 5)$	[1, 0, 0]	[0, 1, 0]	Figure 4.4(g)
$(10, 11)$	[0, 1, 0]	[0, 0, 0]	Figure 4.4(h)

Table 4.2: Sites added by Procedure 2 for the 16×16 test problem

For this test problem, the optimal objective value of the integer model (P_2) when $C = 3$ is 56003483.86. So it is clear that the feasible solution generated by Procedure 2 is not optimal. Thus in general we attempt to improve the solution given by Procedure 2 by using Procedure 3.

To illustrate, we use the feasible solution in Figure 4.4(h) given by Procedure 2 as the starting point for Procedure 3. Figure 4.4(h) shows that there are 23 and 21 boundary sites in the upper left cluster and in the lower right cluster respectively. So the set SE contains 44 sites. Also Figure 4.4(h) shows that there are 15 available sites adjacent to the boundary sites in the upper left cluster and 24 sites adjacent to the boundary sites in the lower right cluster. Here it should be mentioned that we do not consider as available the sites on the outside border of the grid. Therefore

the set NS contains 39 sites.

For these sets SE and NS, application of the improvement algorithm identifies all feasible movements that can be used to improve the feasible solution given by Procedure 2. Table 4.3 contains relevant information about these feasible movements. The first and second columns of Table 4.3 indicate the sites u and v that can be interchanged to improve the current solution. The third and fourth columns show the changes in boundary length ΔBL_{uv} and the changes in distance ΔDIS_{uv} if we interchange the sites u and v. For example if we remove the site $u = (10, 8)$ from its adjacent cluster and add the site $v = (12, 6)$ to its adjacent cluster, the movement will reduce the boundary length by 4 and the overall distance by 0.10.

u	v	ΔBL_{uv}	ΔDIS_{uv}
$(10, 8)$	$(12, 6)$	4	0.10
$(10, 8)$	$(13, 6)$	4	0.99
$(10, 11)$	$(12, 6)$	4	14.56
$(10, 11)$	$(13, 6)$	4	15.13
$(11, 6)$	$(12, 6)$	2	9.37
$(11, 6)$	$(13, 6)$	4	10.50
$(12, 5)$	$(10, 10)$	2	30.68
$(12, 5)$	$(10, 12)$	2	3.44
$(12, 5)$	$(15, 9)$	2	32.45
$(12, 5)$	$(15, 11)$	2	18.84

Table 4.3: Feasible movements and effects on BL and DIS for the first iteration

Based on the data in Table 4.3, we construct the bipartite graph H shown in Figure 4.5, where $|U| = 4$ and $|V| = 6$. As seen in Table 4.3, each feasible move m_{uv} will reduce the boundary length ΔBL_{uv} by 4 or 2. The change in distance values ΔDIS_{uv} associated with each feasible move is shown in the last column of Table 4.3. We apply a maximum weight matching algorithm to the bipartite network H with edge weights ΔBL_{uv}. There are four maximum weight bipartite matchings for this case and we select the matching with maximum reduction in distance. It is shown in Figure 4.5 using dashed edges.

Figure 4.6 shows the interchanges of sites according to this solution of the maximum matching problem. Figure 4.6(a) shows the interchange of site $(10, 11)$ in Figure 4.4(h) with site $(12, 6)$, which individually reduces the total boundary length by 4 and the total distance by 14.56 (row 3 of Table 4.3). Figure 4.6(b) shows that interchanging site $(11, 6)$ in Figure 4.4(h) with site $(13, 6)$ will individually reduce the total boundary length by 4 and the total distance by 10.50 (row 6 of Table

4.3). By making both of these feasible movements the resulting solution has boundary length 58 and total distance 3533.60. So we have decreased the boundary length by 8 and the total distance by 25.82 compared to the feasible solution given by Procedure 2 (Figure 4.4(h)). Thus Procedure 3 has improved the solution given by Procedure 2.

Since there may be further improvements possible, we apply Procedure 3 again to the solution shown in Figure 4.6(b). For the solution in Figure 4.6(b) there are 42 sites in the set SE and 37 in the set NS. With these sets SE and NS, a second application of the improvement algorithm identifies all feasible movements that can be used to improve the feasible solution in Figure 4.6(b). Table 4.4 contains relevant information about these feasible movements. As before the first and second columns of Table 4.4 indicate the sites u and v that can be interchanged to improve the current solution. The third and fourth columns show the changes in boundary length ΔBL_{uv} and the changes in distance ΔDIS_{uv} if we interchange the sites u and v. Now based on the entries of Table 4.4 we see that only one feasible move m_{uv} reduces the boundary length but this feasible move slightly increases the distance. Figure 4.8 shows that interchanging site $(10, 8)$ in Figure 4.6(b) with site $(11, 6)$ will individually reduce the total boundary length by 2 and will individually increase the total distance by 5.57 (row 10 of Table 4.4). A maximum weight bipartite matching for this case is shown in Figure 4.7 using dashed edges.

u	v	ΔBL_{uv}	ΔDIS_{uv}
$(2, 2)$	$(15, 9)$	0	29.37
$(2, 2)$	$(15, 11)$	0	11.70
$(5, 7)$	$(15, 9)$	0	19.30
$(5, 7)$	$(15, 11)$	0	1.63
$(9, 4)$	$(15, 9)$	0	53.38
$(9, 4)$	$(15, 11)$	0	35.71
$(10, 8)$	$(3, 8)$	0	-47.25
$(10, 8)$	$(10, 2)$	0	-91.45
$(10, 8)$	$(10, 11)$	0	-13.82
$(10, 8)$	$(11, 6)$	2	-5.57

Table 4.4: Feasible movements and effects on BL and DIS for the second iteration

By making this feasible movement the resulting solution has boundary length 56 and total distance 3539.17. So we have decreased the boundary length by 2 and increased the total distance by 05.57 compared to the improved solution given by the first iteration. The improved feasible solution is shown in Figure 4.8.

We now apply Procedure 3 again to the solution shown in Figure 4.8. For this solution there are 42 sites in the set SE and 37 in the set NS. With these sets SE and NS, a third application of the improvement algorithm identifies all feasible movements that can be used to improve the current solution in Figure 4.8. Table 4.5 contains relevant information about these feasible movements. As before the first and second columns of Table 4.5 indicate the sites u and v that can be interchanged to improve the current solution. The third and fourth columns show the changes in boundary length ΔBL_{uv} and the changes in distance ΔDIS_{uv} if we interchange the sites u and v. As seen in Table 4.5 no feasible move m_{uv} can reduce the boundary length. Thus based on the entries of Table 4.5 we construct the bipartite network H using the edge weights ΔDIS_{uv}. A maximum weight bipartite matching is shown in Figure 4.9 using dashed lines.

u	v	ΔBL_{uv}	ΔDIS_{uv}
$(2,2)$	$(15,9)$	0	29.47
$(2,2)$	$(15,11)$	0	11.13
$(5,7)$	$(15,9)$	0	19.14
$(5,7)$	$(15,11)$	0	1.06
$(9,4)$	$(15,9)$	0	53.48
$(9,4)$	$(15,11)$	0	35.14

Table 4.5: Feasible movements and effects on BL and DIS for the third iteration

Figure 4.10 shows the resulting interchanges of sites according to this solution of the maximum matching problem. Figure 4.10(a) shows the interchange of site $(9,4)$ in Figure 4.8 with site $(15,9)$, which individually reduces the total distance by 53.48. Figure 4.10(b) shows that interchanging site $(2,2)$ in Figure 4.8 with site $(15,11)$ will individually reduce the total distance by 11.13. By making both of these feasible movements the resulting solution in Figure 4.10(b) has boundary length 56 and total distance 3483.86. So we have decreased the total distance by 55.31 compared to the feasible solution in Figure 4.8. In fact the solution in Figure 4.10(b) is exactly the optimal integer solution to this particular problem and we can stop applying Procedure 3.

In general when model (P_2) does not find a feasible solution, this three-stage heuristic should give good approximate solutions. Occasionally these may be optimal but in general we expect them to be feasible solutions that are reasonably close to the optimal solution of model (P_2).

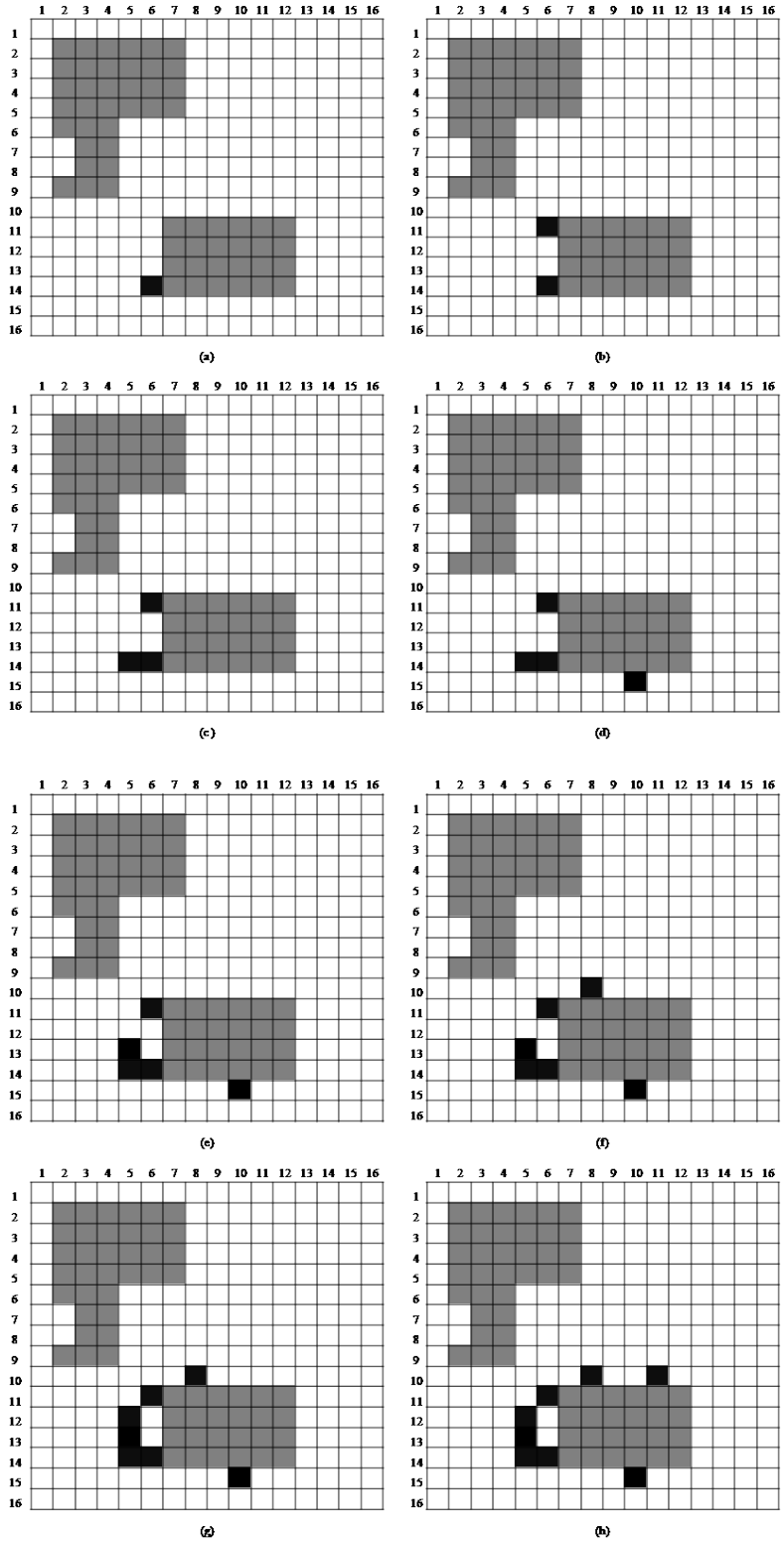

Figure 4.4: Illustration of applying heuristic Procedure 2 to the 16×16 test problem

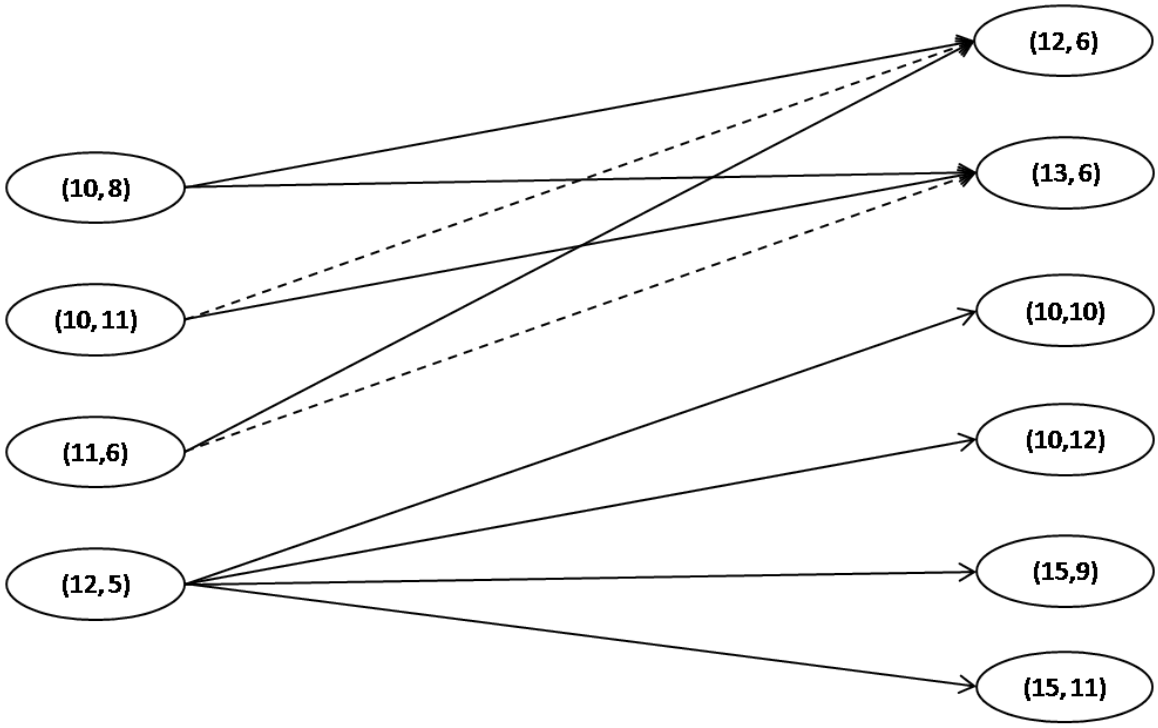

Figure 4.5: Bipartite graph used by Procedure 3 for the 16×16 grid for the first iteration

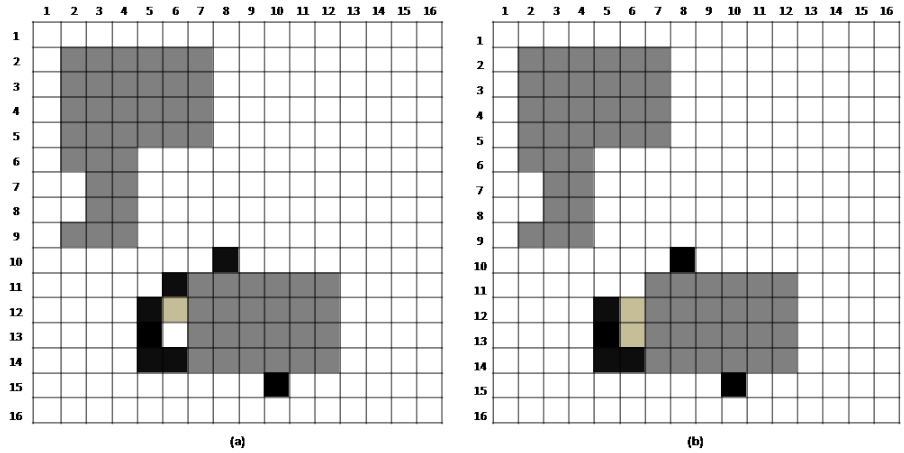

Figure 4.6: Applying Procedure 3 to the 16×16 test problem for the first iteration

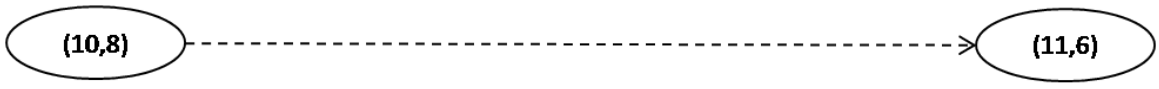

Figure 4.7: Bipartite graph used by Procedure 3 for the 16×16 grid for the second iteration

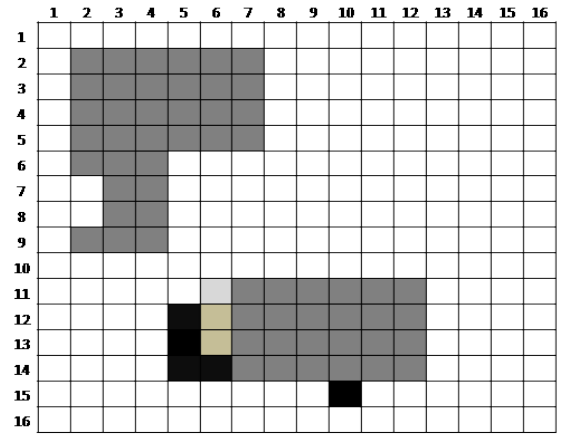

Figure 4.8: Applying Procedure 3 to the 16 × 16 test problem for the second iteration

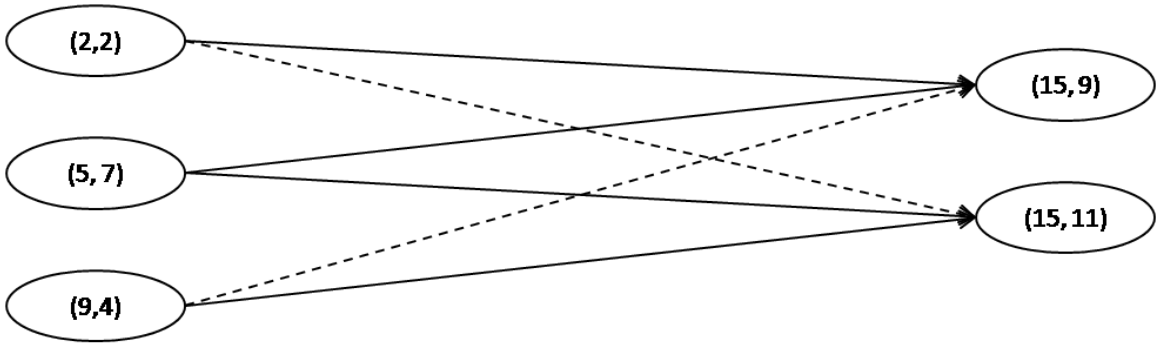

Figure 4.9: Bipartite graph used by Procedure 3 for the 16 × 16 grid for the third iteration

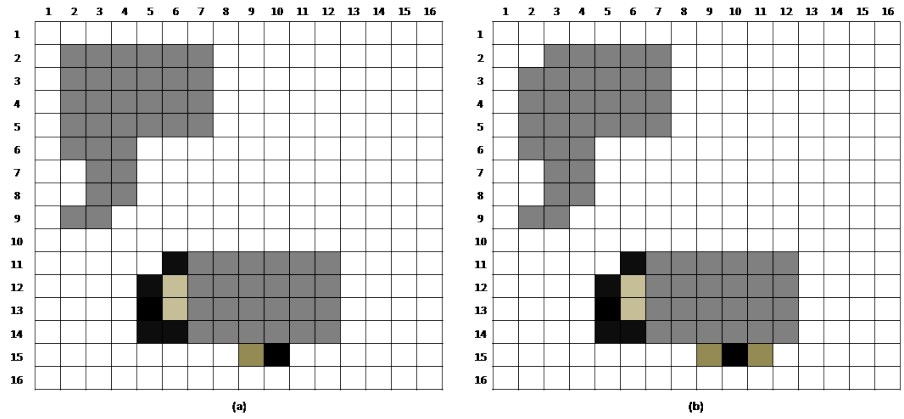

Figure 4.10: Applying Procedure 3 to the 16 × 16 test problem for the third iteration

Chapter 5

Conclusions and Future Research

In the introduction we discussed the importance of designing spatially compact reserve systems. The main limitation of the SCP and MCP approaches is that these two formulations do not consider the spatial relationships among sites selected for the reserve system. Therefore, by taking such considerations into account we wish to cluster reserve sites into a relatively small number of compact groups called *clusters*. To create compact clusters we considered two factors: minimizing the boundary length of all selected clusters and minimizing the total distance between all pairs of sites within each cluster. In order to minimize a suitable combination of these two factors we used a weighting technique to combine these multiple objectives into a single objective function. Also we argued that when creating compact clusters the boundary length of the selected clusters is more important than the distance between all pairs of sites within each cluster. Therefore an appropriate weight U was calculated to give priority to minimizing the boundary length.

In Chapter 2 we introduced our mathematical models and explained each term of the models in detail. Also we explained the calculation of the upper bound U based on the Wiener index. As mentioned in the introduction our initial formulation is a nonlinear integer programming problem and therefore solving the model exactly is time consuming. To solve the model more efficiently we converted the initial model into a linear model. In Section 2.4 we verified that the linear mathematical model did not change the optimal solution of the quadratic model. Therefore we used model (P_2) in all our subsequent numerical investigations since it requires less computational time.

In Chapter 3 we presented our computational experience with the above models using various data sets. We discussed how the optimal solutions behave when the allowable number of clusters

C is changed. Then we showed that $C = 2$ provides a clustering with the optimal boundary length although it may not provide the optimal distance within clusters. Since our first priority is minimizing the boundary length of the selected clusters we chose to use $C = 2$ because it provides reasonable clusters. This was useful in reducing the required CPU time significantly.

Although exact algorithms can identify an optimal solution of model (P_2), for large reserve design problems it is difficult to find an optimal solution in a reasonable amount of time. Consequently, we developed a heuristic procedure in Chapter 4, which is based on a linear relaxation of model (P_2). By using a numerical example we illustrated our three-step heuristic procedure. We noticed that the heuristic solution and the optimal integer solution to model (P_2) for this numerical example were the same. In general we expect this heuristic procedure to obtain good feasible solutions in cases where solving exactly the mixed integer model (P_2) would not be possible.

There are several directions for future research on this topic. We developed a heuristic procedure based on the relaxed model (P_2). We started with sites (i, j) with $X_{cij} = 1\,(c > 1)$ in the relaxed optimal solution.

1. **Improve Procedure 1**: It is possible to create several initial solutions from the relaxed solution by using sites (i, j) with sufficiently large values for X_{cij}.

2. **Improve Procedure 2**: To enhance the quality of the first feasible solution, it is possible to consider more factors in Procedure 2.

3. **Improve Procedure 3**: We can add more sophisticated interchange strategies to obtain good feasible solutions.

Moreover, it will be useful to carry out additional and more extensive testing of the heuristic approach on large test problems.

Bibliography

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. Network flows. Prentice-Hall, New Jersey, 1993.

[2] Michael A. Clemens, Charles S. ReVelle, and Justin C. Williams. Reserve design for species preservation. *European Journal of Operational Research*, 112:273–283, 1999.

[3] Matthias Ergott. Multicriteria optimization. Springer, second edition, 2005.

[4] Douglas T. Fischer and Richard L. Church. Clustering and compactness in reserve site selection: An extension of the biodiversity management area selection model. *Forest Science*, 49:555–565, 2003.

[5] Douglas T. Fischer and Richard L. Church. The SITES reserve selection system: A critical review. *Environmental Modeling and Assessment*, 10:215–228, 2005.

[6] Ante Graovac and Tomaz Pisanski. On the Wiener index of a graph. *Journal of Mathematical Chemistry*, 8:53–62, 1991.

[7] John Hof and Curtis H. Flather. Accounting for connectivity and spatial correlation in the optimal placement of wildlife habitat. *Ecological Modeling*, 88:143–155, 1996.

[8] Rex K. Kincaid, Catherine Easterling, and Meagan Jeske. Computational experiments with heuristics for two nature reserve site selection problems. *Computers and Operations Research*, 35:499–512, 2006.

[9] Mark D. McDonnell, Hugh P. Possingham, Ian R. Ball, and Elizabeth A. Cousins. Mathematical methods for spatially cohesive reserve design. *Environmental Modeling and Assessment*, 7:107–114, 2002.

[10] Darek J. Nalle, Jeffrey L. Arthur, Claire A. Montgomery, and John Sessions. Economic and spatial impacts of an existing reserve network on future augmentation. *Environmental Modeling and Assessment*, 7:99–105, 2002.

[11] Darek J. Nalle, Jeffrey L. Arthur, and John Sessions. Designing compact and contiguous reserve networks with a hybrid heuristic algorithm. *Forest Science*, 48:59–68, 2002.

[12] Hayri Onal. First-best, second-best, and heuristic solutions in conservation reserve site selection. *Biological Conservation*, 115:55–62, 2003.

[13] Hayri Onal and Robert A. Briers. Incorporating spatial criteria in optimum reserve network selection. *Proceedings of the Royal Society of London, Biological Sciences*, 269:2437–2441, 2002.

[14] Hayri Onal and Robert A. Briers. Designing a conservation reserve network with minimal fragmentation: A linear integer programming approach. *Environmental Modeling and Assessment*, 10:193–202, 2005.

[15] Hayri Onal and Yicheng Wang. A graph theory approach for designing conservation reserve networks with minimal fragmentation. *Networks*, 51:142–152, 2008.

[16] Stuart L. Pimm and John H. Lawton. Planning for biodiversity. *Science*, 279:2068–2069, 1998.

[17] Charles S. ReVelle, Justin C. Williams, and John J. Boland. Counterpart models in facility location science and reserve selection science. *Environmental Modeling and Assessment*, 7:71–80, 2002.

[18] Takeshi Shirabe. A model of contiguity for spatial unit allocation. *Geographical Analysis*, 37:2–16, 2005.